НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«На правах рукопису»
УДК 004.91

«До захисту допущено»
Науковий керівник кафедри
_____ І.А. Дичка
«___»_____2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 Інженерія програмного забезпечення

на тему: «Модифікований метод острівної кластеризації природномовних текстових даних»

Виконав:
студент VI курсу, групи КП-61м
Юсин Яків Олексійович _____

Керівник:
Доцент кафедри ПЗКС, к.т.н., доцент,
Заболотня Т.М. _____

Рецензент:
Доцент кафедри ММСА ІПСА, к.т.н., доцент,
Дідковська М.В. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2018 року

ЗМІСТ

ВСТУП

Починаючи з 1950-х років кількість інформації, що генерується людством невпинно зростає. Це явище отримало назву інформаційного вибуху (за даними оксфордського словника цей термін був вперше використаний в 1964 році) [1], і в першу чергу стосується текстових документів, що використовуються в науці, бізнесі і інших галузях діяльності.

Відповідно до досліджень, лише за 2012 рік людство згенерувало 2.8 зетабайти інформації, і за прогнозами ця кількість буде подвоюватись кожні два роки. Проте з цих даних потенційно корисними є лише 23%, а структурованими – лише 5% [2].

Наслідком такого росту об’єму текстових даних стала необхідність в розробленні методів автоматичної попередньої систематизації цих масивів даних. Крім цього, без систематизації текстів неможливе вирішення і інших задач, таких як:

- автоматичне реферування текстових корпусів;
- визначення взаємозв’язків між документами;
- наглядна візуалізація колекцій;
- пошук дублікатів і т.д.

В таких умовах розроблення методів кластеризації та класифікації (які є подібними задачами) текстових документів, тобто розбиття корпусів документів на підмножини (в випадку кластеризації не заданих наперед кластерів, в випадку класифікації – на попередньо задані класи) становиться важливою задачею [3]. Смислова кластеризація (яка і розглядається в даній роботі) передбачає розбиття текстового корпусу на такі кластери, що тексти в межах одного кластеру максимально подібні по змісту між собою, а в межах різних кластерів розташовуються тексти з максимально відмінним змістом між собою. Часто, при подальшому

аналізі отриманого результату, експерт легко може визначити тематичний зміст кожного кластеру.

До 1980-х років основними методами кластеризації текстів були ручні методи, які базуються на використанні оцінок експертів для визначення тематики документів. На той час це було єдиною парадигмою, що дозволяла розроблювати методи кластеризації. Сьогодні цей підхід залишається ефективним для вирішення задач, де необхідне володіння спеціальною інформацією щодо відношення того чи іншого тексту до певного кластеру. Класичним прикладом такої задачі є кластеризація історій хвороб пацієнтів в БД лікарень, оскільки для такої кластеризації необхідна медична освіта для врахування всіх факторів. Але є очевидним те, що ручні методи кластеризації мають недоліки, які є неприйнятними для сучасних умов, описаних вище. Такі методи ручної кластеризації можливо застосувати лише для відносно невеликих корпусів документів та потребують багато часу для роботи експерта чи групи експертів.

Основним поштовхом в розробленні методів автоматичної кластеризації текстів стала робота групи Корнельского університету, яка в 1975 році опублікувала статтю [4], що пропонувала описувати тексти у вигляді векторів багатовимірного простору. Один із головних результатів цієї роботи – модель VSM (Vector Space Model) – описано в розділі 1.

Всі ці описані факти (різке зростання об’єму інформації, різкий розвиток інформаційних та комп’ютерних технологій, неможливість використання в таких умовах ручної кластеризації) вказують на те, що дослідження в області розроблення нових методів кластеризації текстів та підвищення ефективності існуючих методів є актуальною задачею.

1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ КЛАСТЕРИЗАЦІЇ ПРИРОДНОМОВНИХ ТЕКСТОВИХ ДАНИХ

1.1. Задача кластеризації текстових колекцій

Введемо деякі визначення, які будемо використовувати надалі для опису поняття кластеризації і формального визначення задачі кластеризації природномовних текстових даних, загальний опис якої був наведений в вступі.

Визначення 1. Під кластеризацією корпусу текстових документів мається на увазі смислова кластеризація, тобто визначення наявності і складу тематично подібних груп в корпусі документів в випадку, коли апріорний опис цих груп відсутній [5].

Визначення 2. Кластеризація корпусу текстових документів називається автоматичною, якщо всі рішення про склад кластерів приймаються автоматично ЕОМ, без участі людини.

Визначення 3. Задачею кластеризації корпусу текстових документів називається наступна задача: нехай задано корпус текстових документів X. Необхідно розбити корпус на підмножини (кластери) так, щоб кожен кластер містив тексти подібні за змістом, а тексти різних кластерів відрізнялися за змістом. Результатом виконання цієї задачі є зіставлення кожному тексту корпусу мітки кластеру (наприклад, номеру).

В випадку деяких прикладних задач визначення 3 може бути модифікованим для дозволу розбиття корпусу X на кластери, що пересікаються (так звана неексклюзивна кластеризація), в такому випадку кожному тексту корпусу зіставляється множина міток кластерів.

Для автоматичної кластеризації корпусу текстових документів будь-який метод оперує способом визначення близькості текстів за змістом. Існують різні можливості для цього:

– близькість документів за змістом може бути визначена на основі підрахунку кількості термів з деякого заданого набору, які входять в тексти;
– якщо тексти розглядаються як точки багатовимірного простору, близькість документів може бути визначена за допомогою введення метрики близькості;
– можливо визначити близькість документів на основі кореляції термів, що складають ці тексти і т.д..

Якщо використовується метрика близькості, постановка задачі кластеризації корпусу текстових документів може бути уточнена.

Визначення 4. Задачею кластеризації корпусу текстових документів називається наступна задача: нехай задано корпус текстових документів X і метрику близькості $\mu(x_1, x_2)$, де документ $x_i \in X$. Необхідно розбити множину X на підмножини $\{X_i\}, \bigcup_i X_i = X$, так, щоб кількість різних трійок текстів $x_1, x_2, x_3 \in X$ таких, що $x_1 \in X_i; x_2, x_3 \in X_j, i \neq j$, а $\mu(x_1, x_2) < \mu(x_2, x_3)$, була найменшою. При цьому кожному документу зіставляється мітка його кластеру.

Навіть використання цього визначення і вибір оптимального рішення не зменшує неоднозначність результату кластеризації, що є очевидним – можливо визначити безкінечну кількість різноманітних метрик близькості текстів, які будуть давати різноманітні оптимальні рішення. Таким чином, множина можливих рішень задачі кластеризації безпосередньо залежить від методу кластеризації, що використовується, і різні методи дадуть різні оптимальні результати. Це також пов’язано з тим, що не існує однозначно найкращого критерію якості отриманого результату. В цьому аспекті задача кластеризації істотно відрізняється від задачі класифікації, для якої відома велика кількість критеріїв оцінки отриманого результату. Питання оцінки якості кластеризації більш детально розглянуто в розділі 4. Крім цього, при достатньо великому об’ємі текстового корпусу, від вибору

методу кластеризації суттєво залежить час та ефективність роботи відповідного алгоритму.

Визначення 5. Алгоритмом кластеризації називається відображення $f: X \to \{X_i\}$, яке будь-якому документу $x \in X$ зіставляє у відповідність мітку кластеру, або послідовність кроків по знаходженню образу f.

Мета кластеризації корпусу текстів може бути різною в залежності від особливостей конкретної задачі. Наступні типи задач кластеризації є найбільш розповсюдженими:

- зрозуміти структуру корпусу документів, розбивши його на кластери і оброблювати надалі кожен кластер окремо – в такому випадку фінальна кількість кластерів зазвичай невідома і знаходиться методом автоматично або встановлюється вручну відповідно до суб'єктивних критеріїв;
- зменшити об'єм даних за допомогою збереження тільки одного представника (найбільш «типового») кожного кластеру;
- виділити нетипові тексти, які не входять до жодного кластеру (або іншими словами входять до кластерів, які складаються з них самих).

В зв'язку з неоднозначністю та суб'єктивністю процесу кластеризації корпусу текстових документів, в наступних пунктах розглянуто основні методи кластеризації корпусу документів і визначено вимоги до розроблюваного модифікованого методу в рамках даної роботи.

Також кластеризація текстового корпусу може проводитись одним із двох способів: пласким та ієрархічним.

Пласка кластеризація – найпростіший вид кластеризації. При пласкій кластеризації корпус кожний документ зв'язується лише з однією групою текстів. В більшості випадків саме пласка кластеризація є основним способом кластеризації текстових масивів. Другими назвами цього способу є декомпозиція, розподілення, k-кластеризація.

Ієрархічна кластеризація – вид кластеризації текстів, при якій більш великі кластери, що виділяються на першому етапі, розділяються надалі на більш дрібні. Цей процес повторюється для кожного із отриманих кластерів до виконання деякого критерію зупинки.

Задачі ієрархічної кластеризації множини текстів визначеної предметної області називаються задачами таксономії. Результатом таксономії є деревоподібна структура, яка будується над текстами. Замість однієї мітки кластеру кожний документ характеризується перерахуванням міток всіх кластерів, до яких він належить [6,7].

Також методи кластеризації поділяють на три групи за напрямком виконання кластеризації множини документів:

- зверху-вниз;
- знизу-вверх;
- ті, що не використовують в явному вигляді два способи, описані вище.

До групи «зверху-вниз» відносяться ітеративні процедури, які на кожному кроці розбивають кожний наявний кластер на пару більш дрібних, до тих пір, поки не спрацює критерій зупинки. На першому кроці весь корпус документів розглядається як єдиний кластер і подається на вхід методу кластеризації. Далі він поділяється на два кластери, один з них обирається за певним критерієм і знову ділиться і т.д.. Як правило, критерієм зупинки в такому випадку є кількість отриманих кластерів на певному кроці.

В групі «знизу-вверх» розглядаються ітеративні процедури, які на кожному кроці об’єднують найбільш близькі кластери до тих пір, поки не виконається критерій зупинки. На першому кроці кожний документ розглядається як окремий кластер. Далі об’єднуються два найбільш близькі кластери, далі ще два і т.д.. Цей тип методів кластеризації також називають агломеративним.

1.2. Огляд основних методів кластеризації природномовних текстових даних

На сьогодні більшість методів кластеризації текстових документів використовують векторну модель (VSM), яка згадувалася в вступі.

Нехай задано корпус текстів і T – це множина всіх різноманітних термів, з яких складаються документи (після нормалізації), а N – кількість термів в цій множині (в деяких модифікаціях N це кількість термів всієї мови, на якій написані тексти).

В такому випадку кожний документ корпусу може бути представлений у вигляді вектору довжини N, координати якого відповідають термам множини T, і мають значення 1, якщо цей терм зустрічається в тексті, 0 – якщо ні. Таким чином, у векторній моделі текст розглядається як сукупність термів, що його складають (такий підхід отримав назву bag of words).

Є очевидним те, що модель VSM представляє корпус документів у вигляді підмножини всіх можливих текстів, які можливо побудувати з доступних термів. Введення додаткових лінійних структур над такою множиною дозволяє розглядати його в якості множини векторів в багатовимірному просторі.

Існують і інші модифікації моделі VSM. Поширеною є модифікація, яка в якості векторного простору розглядає простір $\mathbb{R}^n$, а в якості значень для координат векторів, що відповідають текстам, використовується або кількість входжень, або частоти. Таким чином, простір векторів перетворюється в неперервний, а не дискретний. Цей підхід дозволяє використовувати інформацію про частотний розподіл термів в корпусів тексті вже на етапі формування векторного простору [8].

Також, наприклад, було розроблено модифіковану векторну модель для визначення характеристик текстів на основі визначених наперед характеристик структурних шматків цих текстів [9].

Основний недолік моделі VSM полягає в тому, що в якості координат для узагальнених векторів текстів використовуються насправді залежні змінні: наявність в документі деякого набору термів вже передбачає велику ймовірність наявності в ньому інших термів, що пов’язані з цим набором. В реальних випадках незалежність термів можливо встановити, розглянувши розподіл різних термів по корпусу. Використовуючи цей спосіб, можливо вибрати для координат в моделі ті терми, які між собою мало корелюють [10]. Незважаючи на цей недолік, векторна модель залишається найбільш опрацьованою і використовуваною моделлю подання текстів.

Більшість методів мають основні загальні кроки, такі як приведення термів до початкової форми та зменшення потужності множини термів.

Для того, щоб різноманітні форми одного терму не розглядались як не зв’язані один з одним, всі терми, що входять в корпус документів, приводяться до нормальних форм (це також зменшує потужність множини термів).

На сьогодні можливо виділити три основні розроблені методи морфологічного аналізу тексту (які отримали назву стеммінгу [11]):

- метод таблиць закінчень;
- метод правил словотворення;
- словниковий метод.

Метод таблиць закінчень [11] передбачає створення та використання списку всіх можливих закінчень в мові, що розглядається. Кожне виділене із документу слово ставиться в відповідність гаданій початковій формі цього слова, яка отримується відокремленням від слова максимально довгого закінчення цього слова, що міститься в таблиці.

Перевагою цього методу є висока швидкість роботи (в більшості мов потужність множини можливих закінчень не перевищує декількох сотень, наприклад, для англійської мови це число 250 [11]). Недоліком цього методу є велика кількість помилок.

Метод правил словотворення передбачає фіксацію набора правил побудови форм слів мови. Кожне слово методом оберненого ходу перевіряється на можливість віднесення до результатів дії деякого правила. В найбільш простому випадку далі вибирається така початкова форма слова, для якої правило, що створює цю форму, має найменшу кількість кроків [11].

Перевагами цього методу є висока швидкодія і мала кількість помилок, а недоліком є неможливість врахування виключень.

Для роботи з виключеннями призначений словниковий метод стеммінгу, який передбачає використання вже існуючого словника форм та пошук по ньому [11].

Для зменшення потужності множини термів застосовуються наступні засоби:

- використання стоп-списків, які містять не змістовні слова;
- використання методів лінгвістики;
- використання словників та тезаурусів для об’єднання нормальних форм і синонімічних груп.

Зменшення набору ознак в найбільшій мірі використовується для зменшення часу роботи кластеризації і використовуваних під час неї ресурсів, що є актуальним при обробленні великого об’єму інформації. Частіше за все зменшення кількості термів знижує якість кластеризації. Незважаючи на це, в випадку однорідних (тобто таких, що містять подібні набори термів) текстів в корпусі можна досягнути лише невеликого зниження якості при суттєвому зменшенні кількості термів, і в окремих випадках навіть отримати вищу якість.

Основні методи кластеризації, які розглянуто надалі, можливо розділити по тому критерію, чи використовуються для кластеризації нечислові характеристики документів, відмінні від їх математичних (статистичних) властивостей.

Відповідно до такої класифікації, при розгляді в загальному вигляді, методи кластеризації можуть бути розділені на дві групи:

- методи, які розглядають тексти у вигляді узагальнених векторів, відповідно до моделі VSM, і не використовують нечислові характеристики текстів;
- методи, які використовують нечислові характеристики документів.

Методи, які будуть розглянуті надалі в цьому пункті, у відповідності до цієї класифікації розподіляються чином, зображеним на рис. 1.1.

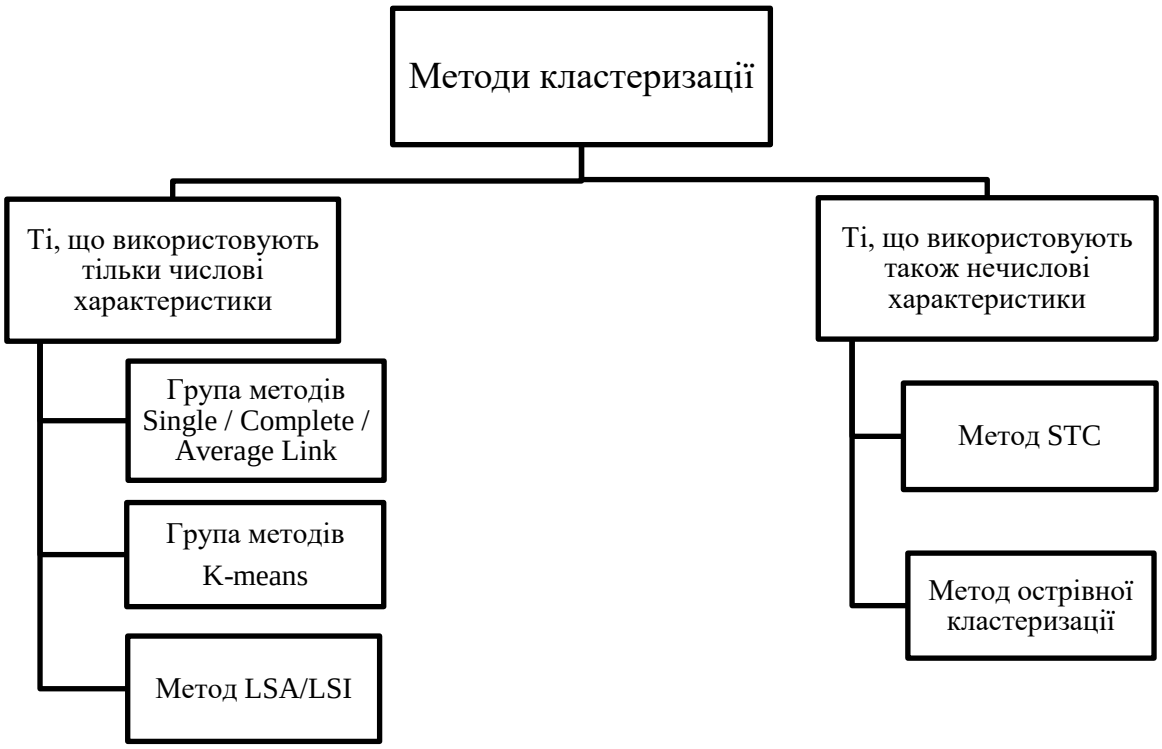

Рис. 1.1. Класифікація методів кластеризації

Вибір саме цих методів в складі кожної групи визначений тим, що ці методи найчастіше використовуються для кластеризації – така їх популярність пов’язана з тим, що ці методи включені в різноманітні популярні програмні продукти.

1.2.1. Група методів Single / Complete / Average Link

Найбільш поширеними ієрархічними методами, які засновані на близькості текстів в просторі, є методи Single / Complete / Average Link

[12]. Особливістю цих методів є те, що вони розбивають документи на кластери за допомогою їх розбиття на ієрархічні групи.

Результатом роботи цих методів є бінарне дерево або дендрограма, яка пов’язує всі тексти корпусу, що кластеризується. При заданій вручну кількості кластерів вибір відповідного зрізу бінарного дерева дає розбиття тексту на кластери.

Приклад алгоритму, що реалізує ці методи, складається з таких кроків:

1. знаходяться значення функції близькості між документами і формується матриця близькості;
2. кожний документ відноситься до свого окремого кластеру;
3. найбільш близькі пари документів об’єднуються в один кластер;
4. знайдена матриця близькості оновлюється шляхом видалення строк і стовбців для кластерів, що були об’єднані з іншими;
5. матриця близькості перераховується;
6. перехід на крок 3 до тих пір, поки не спрацює критерій зупинки.

Три методи відрізняються один від одного шляхом визначення відстані між кластерами. Відстань між кластерами визначається так:

- в методі Single Link відстань між кластерами визначається як мінімальна відстань між парою об’єктів в сусідніх кластерах;
- в методі Complete Link – максимальна відстань між парою об’єктів в сусідніх кластерах;
- в методі Average Link – середня відстань між елементами двох кластерів.

Внаслідок використання різних способів підрахунку відстані між кластерами ці три методи мають різну точність. Перевірка точності методів проведена на спеціальних тестових наборах і визначено, що алгоритм Single Link має найменшу точність, а Complete Link та Average Link – приблизно рівні між собою [13].

Що стосується складності алгоритмів, що реалізують дані методи, то складність алгоритму Single Link відповідає $O(N^2)$, а алгоритму Complete Link – $O(N^3)$, де N – це кількість текстів в корпусі. Метод Average Link є компромісом по складності між ними.

Приклад кластеризації цими методами наведено на рис. 1.2.

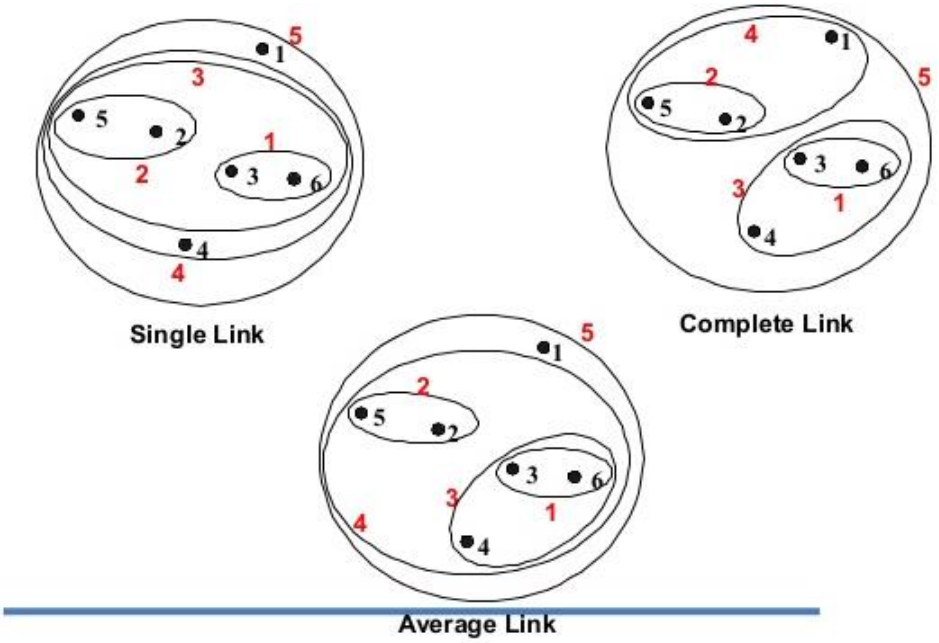

Рис. 1.2. Результат кластеризації методами Single / Complete / Average Link

1.2.2. Група методів k-means

Методи цієї групи є одними із найбільш поширених методів виконання пласкої кластеризації. В найбільш простій реалізації передбачається ручне встановлення числа кластерів і початкових позицій центроїдів кластерів, після чого розпочинається процес, який стабілізує позиції центроїдів [14]. На кожному кроці алгоритму, що реалізує цей метод, документи приписуються до кластеру з центроїдом, відстань до якого є найменшою. Після того, як всі тексти розподілені, знаходяться нові позиції центроїдів. Виконання алгоритму зупиняється, коли центроїди вже не переміщуються, або коли виконано критерій зупинки [14].

Існують різні модифікації базового методу, наприклад такі, в яких для знаходження початкових позицій центроїдів використовуються методи

Single / Average Link на випадковій підмножині [15]. В такій модифікації розмір цієї випадкової вибірки рівний $\sqrt{kN}$, де k це кількість кластерів, а N це кількість текстів.

Також існують модифікації, які автоматизують вибір k – кількості кластерів. Крім цього, відома модифікація методу під назвою k-medoids, в якій центроїдом може бути лише точка з корпусу, що кластеризується. Ця модифікація в першу чергу розроблена для кластеризації графів.

Алгоритм, що реалізує найбільш простий різновид даного методу, складається з наступних кроків:

1. Випадково обираються k текстів з корпусу, які приймаються в якості початкових центроїдів.
2. Всі тексти корпусу розподіляються серед кластерів. Документ може потрапити лише в один кластер, метрика близькості до центроїда якого має найбільше значення.
3. Перераховуються центроїди кластерів, виходячи з нової множини документів в кожному кластері.
4. Якщо центроїди кластерів не перемістились, або виконано критерій зупинки, алгоритм закінчує своє виконання.
5. Інакше алгоритм повертається на п.2.

Приклад кластеризації даним методом наведено на рис. 1.3. Перша частина рисунку зображує початкові точки та випадково обрані центроїди, друга та третя відповідають одному кроку алгоритму, а остання – фінальному результату кластеризації.

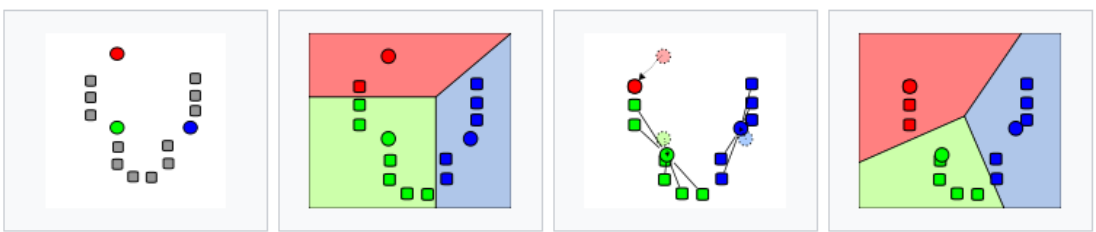

Рис. 1.3. Хід виконання методу k-means

1.2.3. Метод латентного семантичного аналізу LSA/LSI

Цей метод фактично об’єднує в один процес варіант кластеризації на рівні термів (яка використовуються для групування родинних слів) і кластеризацію на рівні документів. Також цей метод знижує розмірність простору ознак за рахунок аналізу кореляції термів.

Метод латентного семантичного аналізу LSA/LSI (це назва одного методу, проте під назвою LSI прийнято позначати його застосування для індексації, а під LSA – для всіх інших областей) враховує зв’язки між елементами векторів tf-idf, тобто використовується спільна зустрічальність на рівні тексту [16]. Ці вектори складаються в матрицю, яка надалі розкладається по базису головних компонент, розмір якого задається вручну. Після цього застосовуються інші методи кластеризації, наприклад розглянуті вище Single/Average Link або k-means.

В основі цієї операції лежить процедура сингулярного розкладання (SVD, singular value decomposition) – розкладання по сингулярним значенням, завдяки якому матрицю TF-IDF можливо представити у вигляді $Q = U * D * V^{-1}$. D – це діагональна матриця сингулярних значень матриці, а U, V – матриці з ортонормованими стовпцями лівих та правих сингулярних векторів відповідно.

Головним недоліком цього методу, незважаючи на ряд переваг, є велика обчислювальна складність алгоритму, що його реалізовує. Ця складність рівна $O(k^3N^2)$, де k – це зменшена розмірність простору ознак.

1.2.4. Метод Suffix Tree Clustering (STC)

Метод STC заснований на використанні суфіксних дерев, що використовуються для швидкого пошуку (за пропорційний довжині пошукового рядку час) [17]. В методі STC будується структура суфіксного дерева для послідовності ідентифікаторів, що об’єднують всі тексти.

Суфіксне дерево – це дерево, що містить всі суфікси даного рядка. Воно складається з вершин, гілок і додаткових вказівників, за допомогою

яких отримують лінійну швидкість побудови дерева. Гілки дерева позначаються літерами чи сполученнями літер, які є частинами суфіксів рядка. Суфікс, який відповідає певній вершині, можливо отримати шляхом об’єднання всіх літер, які розміщені на ребрах дерева, розпочинаючи від кореня і закінчуючи даною вершиною.

Як і в інших методах, при побудові цієї структури можуть використовуватись ідентифікатори словоформ, нормальних форм або синонімічних груп. В кожному вузлі зберігається інформація про документи, які пройшли через цей вузол. Зовнішні вузли відповідають, як правило, окремим словам, а внутрішні – словосполученням, які часто використовуються. Вузли, які відповідають окремим словам або словосполученням, розглядаються як основи для формування кластерів. Основи об’єднуються, якщо містять достатній відсоток спільних текстів. Процес кластеризації зупиняється, коли більше немає кластерів, що сильно пересікаються. При цьому кожний кластер позначається набором слів та словосполучень, які відповідають вузлам початкового дерева, що в нього ввійшли.

Приклад такого суфіксного дерева наведено на рис. 1.4.

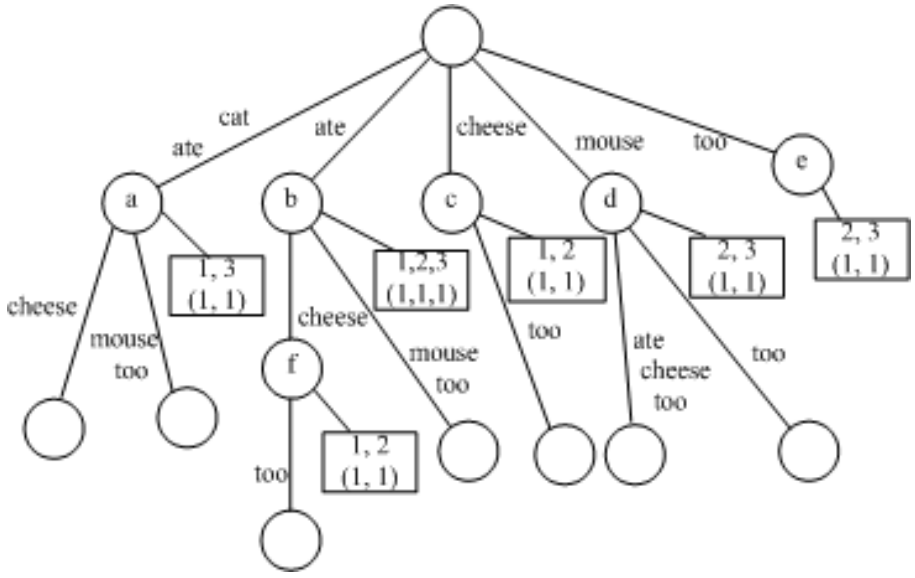

Рис. 1.4. Суфіксне дерево

Цей метод має такі переваги, як:

– висока швидкість роботи – час побудови дерева пропорційний кількості текстів.
– найгірший теоретичний випадок швидкості роботи пропорційний квадрату кількості документів;
– не потребує навчання і завдання порогу спрацьовування.

1.2.5. Метод острівної кластеризації

Це відносно молодий метод, який існує в різних модифікаціях. Основною ідеєю цього методу є підхід, що складається з двох кроків: спочатку цей метод кластеризує терми, з яких складаються документи, і вже на основі отриманих кластерів будуються кластери документів.

В деяких модифікаціях цього методу можуть використовуватись нечислові характеристики текстів та термів. Наприклад, в таких галузях застосування, як лінгвістика, антропологія, етимологічні дослідження, для кластеризації термів можуть використовуватись такі їх характеристики, як час появи слова в мові або місцевість, де він використовувався.

Основним методом острівної кластеризації є метод, який базується на використанні графа сумісної зустрічальності термів. Цей граф будується на основі визначення кореляції кожної пари термів, з яких складаються тексти, і кластеризується саме цей граф (або його частина) описаним в методі підходом. Таким чином, терми документів групуються в кластери саме на основі спільної зустрічальності. Також, завдяки цьому цей метод є стійким до проблем синонімії та омонімії – терми з різним значенням з великою ймовірністю потраплять в різні кластери термів, оскільки вони будуть зустрічатись в текстах спільно з різними термами. В парі до цього для вирішення омонімії можливо використовувати словники.

Вже для групування текстів в кластери на основі кластерів термів використовуються спеціальні процедури, що також описані цим методом.

Таким чином, метод острівної кластеризації текстових колекцій складається з наступних кроків:

1. Попереднє оброблення текстів з вхідної колекції документів: видалення стоп-слів, лематизація тощо.
2. Виділення з текстів множини термів, з яких вони складаються.
3. За необхідності – фільтрація отриманої множини термів (наприклад, в ситуаціях, коли відомі початкові центроїди кластерів або отримана множина є занадто великою).
4. Побудова графу кореляції термів між собою.
5. Попереднє оброблення графу і отримання його наближення.
6. Кластеризація отриманого наближення графу.
7. Розбиття документів на кластери на основі отриманих кластерів термів.

Як правило, цей метод дає кластери, що легко інтерпретувати саме на основі змісту документів, що складають ці кластери.

1.3. Визначення вимог, які визначають ефективність процедури кластеризації

Як вже сказано, оцінювання якості результатів кластеризації (а відповідно і ефективності всієї процедури в цілому) є суб'єктивним процесом, який в першу чергу залежить від вимог, які були поставлені до процедури кластеризації. В свою чергу, ці вимоги часто є залежними від практичної галузі застосування кластеризації. Тому слід визначити вимоги, які будуть поставлені до розроблюваного саме в рамках цього дослідження методу кластеризації текстів.

Головною особливістю процедури кластеризації, яка є спільною для всіх практичних галузей, є вимога інтерпретовності отриманого результату на виході процедури. Результати кластеризації припускають, як правило, їх розуміння та інтерпретацію користувачем методу.

Також з вимоги інтерпретовності випливає інша вимога – вимога ексклюзивності кластеризації. Це означає зв'язок «один-один» між документами та кластерами, тобто один текст може відноситись лише до

одного кластеру. Це пов’язано з тим, що більшість галузей використання процедури кластеризації пов’язана зі строгим відношенням того чи іншого тексту до відповідного кластеру.

Крім вимоги інтерпретовності необхідність практичного застосування методу та практичної застосовності його алгоритму призводить до появи ще декількох вимог до методу кластеризації.

Досвід роботи з алгоритмами оброблення текстів показує те, що для таких алгоритмів прийнятною верхньою межею швидкості роботи є квадратична залежність від кількості текстів [18]. Відповідно, при такій залежності часу роботи алгоритм ще не втрачає своєї працездатності при збільшенні кількості документів в корпусі [18].

Також вимогою до методу кластеризації є бажаність наявності процедур для вирішення синонімії та омонімії в текстах. Це пов’язано з тим, що комп’ютер сам по собі, на відміну від людини, не має здатності швидко визначати контекстуальне значення кожного терму, що є однією з найбільших складностей розвитку математичних методів роботи з текстами.

Часто в більшості практичних застосувань процедури кластеризації вже на початку кластеризації наявні припущення щодо кількості кластерів в корпусі документів. Тому є бажаною наявність можливості вручну встановлювати результуючу кількість кластерів для процедури. При цьому також необхідно залишити можливість автоматичного визначення кількості кластерів.

Ще одна вимога до процедури кластеризації – це її стійкість при розширенні текстового корпусу на більшу підмножину генеральної множини, до якої він належить. Цю вимогу можливо визначити як статистичний характер результатів кластеризації. Ця властивість процедури кластеризації є особливо важливою, якщо розглядати динаміку розподілу документів по кластерам в часі.

Дійсно, якщо користувач, що використовує метод кластеризації, цікавиться результатом кластеризації лише відносно фіксованого набору текстів, який вважається не розширюваним, то можливо розглядати результат не задаючись питанням чи відображає результат властивості більшої сукупності текстів. В протилежність, якщо в задачі розглядається досліджуваний корпус тільки як частина деякої генеральної множини, то природно вимагати від методу кластеризації, щоб отримані результати характеризували всю генеральну множину – і як наслідок, щоб кластери, які будуть отримуватись на різних вибірках цієї генеральної множини, були подібними.

Часто корпус текстів не створюється один раз і надалі використовується без змін, а навпаки, оновлюється через рівні інтервали часу, при цьому зміна корпусу за один такий часовий інтервал є відносно малою (наприклад, задача динаміки новин). Для того, щоб результат кластеризації мав властивість спадковості, тобто стійкості при аналізі таких колекцій, взятих через невеликі інтервали часу, метод кластеризації повинен бути статистичним за своєю природою.

Таким чином, основними властивостями, якими повинен володіти метод кластеризації для того, щоб бути практично застосовним в контексті цієї роботи, є:

- інтерпретовність знайдених кластерів;
- відношення одного документу до одного кластеру;
- час роботи алгоритму кластеризації повинен бути зверху обмеженим квадратичною залежністю від кількості документів в корпусі;
- наявність в методі кластеризації засобів вирішення синонімії та омонімії;
- наявність в методі кластеризації можливості ручного завдання кількості кластерів;

– статистична значимість групування текстів в кластери.

Висновки за першим розділом

На основі огляду найбільш популярних методів кластеризації корпусів текстів та опису вимог, що визначають ефективність методу кластеризації, можливо визначити методи, які можуть бути взяти за основу для побудови модифікованого методу кластеризації природномовних текстових даних.

Порівнявши переваги та недоліки розглянутих методів кластеризації текстів можливо зробити висновок, що поставленим в п.1.3 вимогам в найбільшій мірі відповідає метод острівної кластеризації текстів. Головними його недоліками з точки зору поставлених вимог є те, що цей метод не допускає ручного встановлення очікуваної кількості кластерів та виконання ексклюзивної кластеризації.

Також можливо зробити висновок, що доцільним є проведення детального аналізу кожного етапу даного методу та процедур, що ними використовуються, з метою підвищення якості отримуваних результатів.

Таким чином, метод острівної кластеризації може бути взятим за основу для побудови модифікованого методу кластеризації природномовних текстових даних. Цей модифікований метод повинен прибрати описані недоліки острівної кластеризації і в результаті чого розроблений метод буде повністю задовольняти поставленим вимогам.

Перспективним напрямком для такої модифікації є використання іншого методу на етапі кластеризації графу сумісної зустрічальності термів – такого, який би дозволив ручне встановлення очікуваної кількості кластерів термів та виконання ексклюзивної кластеризації.

Також перспективним напрямком вдосконалення методу острівної кластеризації є розроблення нових підходів до виконання інших етапів методу. В рамках даної роботи буде детально розглянуто етап попереднього оброблення графу сумісної зустрічальності термів.

Отже, для вирішення задачі кластеризації природномовних текстових даних в рамках даної роботи пропонується модифікований метод, що заснований на методі острівної кластеризації, та розширює і модифікує його в частині:

- попереднього оброблення графу сумісної зустрічальності термів;
- кластеризації отриманого наближення графу сумісної зустрічальності термів.

2. МОДИФІКОВАНИЙ МЕТОД ОСТРІВНОЇ КЛАСТЕРИЗАЦІЇ ПРИРОДНОМОВНИХ ТЕКСТОВИХ ДАНИХ

2.1. Підходи до попереднього оброблення графу сумісної зустрічальності термів перед виконанням його кластеризації

Визначимо та опишемо для модифікованого методу кластеризації текстової колекції декілька підходів до попереднього оброблення графу сумісної зустрічальності термів перед виконанням його кластеризації. Метою такого попереднього оброблення є зменшення кількості вершин та ребер графу, що спрощує його оброблення, а також зменшує кількість необхідної пам’яті для його зберігання. На виході процедури оброблення ми отримаємо деяке наближення або апроксимацію вхідного графу, при цьому його досліджуванні характеристики повинні не змінюватись або змінюватись на невелику похибку. В випадку використання графу для кластеризації, такої характеристикою графу є його кластерна структура – процедура попереднього оброблення повинна не змінювати її або змінювати на величину, якою можливо знехтувати.

Є очевидним той факт, що при використанні однакового підходу для кластеризації отриманого наближення графа, саме від якості процедури попереднього оброблення буде залежати якість отриманої кластеризації.

2.1.1. Отримання наближення графу з використанням глобального порогу

Цей підхід використовується оригінальним методом острівної кластеризації. Як зрозуміло з назви підходу, отриманий граф сумісної зустрічальності термів фільтрується за допомогою деякого глобального порогу, і всі ребра, що відповідають кореляції меншій, ніж цей поріг, видаляються з графу. Отримана апроксимація графу використовувалась надалі для її кластеризації.

Позначимо кореляцію термів i та j між собою як p_{ij}, значення якої відповідає вазі відповідного ребра графу сумісної зустрічальності, а значення глобального порогу як $threshold$. Тоді даний підхід можливо реалізувати в наступний спосіб:

1. На вхід алгоритм отримує граф $G = (V, E, w)$.
2. Для кожного ребра графу $e = (i, j)$, порівнюємо його вагу з порогом.
3. Якщо $w_{ij} = p_{ij} \geq threshold$, додаємо ребро e до G_{sparse}.
4. Повертаємо отриманий G_{sparse}.

Оригінальний підхід до оброблення графу сумісної зустрічальності термів, що використовується методом острівної кластеризації, використовує зафіксоване значення $threshold$, яке обчислюється за формулою (2.1) [18].

$$p_c = \frac{0.03}{\max(N_{terms}^2, N_{docs})}, \text{ де } N_{terms} \text{ – кількість всього термів,}$$
$$N_{docs} \text{ – кількість всього текстів в корпусі.} \quad (2.1)$$

Як зазначається авторами, такий вибір дозволяє зменшити кількість малозначимих зв’язків у випадку великого корпусу різноманітних текстів. Також зазначено, що в ході експериментів визначено, що таке значення параметру впливає на склад найбільш малозначимих кластерів, змінюючи їх.

Таким чином, описаний в даному пункті підхід є його узагальненням, що передбачає довільний вибір порогу користувачем.

Описаний підхід є найшвидшим серед описаних в цьому розділі підходів до попереднього оброблення графу. Головний недолік цього підходу до оброблення графу полягає в тому, що він оброблює всі ребра однаково, тобто глобально, що призводить до втрати кластерів термів. Проілюструвати даний недолік і те, як він впливає на якість результату кластеризації, можна за допомогою такого прикладу. Нехай маємо частину графу сумісної зустрічальності термів, що зображена на рис. 2.1. Ребра, що

відповідають відсутній кореляції термів між собою, не показані; суцільним зображено ребра, вага котрих менше значення $threshold$. Ребра, вага котрих дорівнює $1.1 * threshold$, зображено штриховою лінією.

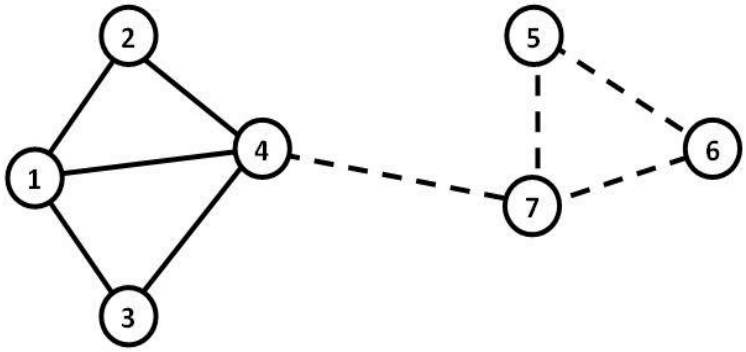

Рис. 2.1. Частина графу сумісної зустрічальності термів

Після виконання описаного підходу оброблення графу отримане його наближення буде містити лише один кластер, що складається з термів №№1-4, при цьому кластер термів №№5-7 буде втрачено (і відповідний йому кластер документів). Якщо ж навпаки, почати збільшувати обране значення порогу, також можливо зменшити якість кластеризації, включивши до апроксимації помилкові ребра.

2.1.2. Отримання наближення графу з використанням відсоткового порогу

Цей підхід можливо розглядати як модифікацію попереднього підходу, який замість наперед визначеного значення глобального порогу використовує деяке, визначене з самого графу.

В ході виконання цього підходу відбувається сортування всіх ребер графу за їх вагою. Як результат, в апроксимацію вхідного графу потрапляє визначений відсоток ребер з найбільшою кореляцією.

Якщо ми позначимо значення відсоткового порогу як s, тоді даний підхід матиме таку реалізацію:

1. На вхід алгоритм отримує граф $G = (V, E, w)$.
2. Відсортуємо всі ребра в E за їх вагою w_{ij}.

3. Додаємо верхні $s\%$ ребер отриманого відсортованого списку до G_{sparse}.

4. Повертаємо отриманий G_{sparse}.

Проведені експерименти на різноманітних текстових колекціях показали, що значення p_c розглянутого першого підходу, відповідає діапазону [3,7] значень s. Таким чином, при використанні більшого значення, ніж права границя цього діапазону, в наближення графу потрапить більше ребер, ніж при використанні першого підходу. Це також призведе до деякого збільшення якості отриманої апроксимації.

Описаний підхід потребує більше часу та більшої кількості операцій, ніж перший, і є по своїй суті двопрохідним. Перший прохід – це сортування всіх ребер, другий – прохід по частині отриманого списку для додавання ребер до апроксимації.

2.1.3. Отримання наближення графу з використанням ефективного опору

В роботі [19] пропонується підхід до апроксимації графу, що заснований на використанні ефективного опору.

Саме поняття ефективного опору для графу будується на основі розгляду цього графу як такого, що представляє собою електричне коло [20]. Ефективний опір R_{ij} між парою вершин графу i та j – це електричний опір в колі, виміряний між вершинами i та j, де ребра графу це резистори з електричною провідністю, що відповідають вазі ребра w_{ij}. Це так званий електричний сенс цієї метрики. Також визначають броунівський сенс – в такому випадку R_{ij} це величина, що пропорційна середньому часу досяжності вершини j з вершини i при випадковому блуканні по графу.

Цей підхід можливо описати наступним алгоритмом [19]:

1. На вхід алгоритм отримує граф $G = (V, E, w)$.

2. Вибираємо випадкове ребро e з графу G з ймовірністю p_e, що пропорційна величині $w_{ij}R_{ij}$.
3. Додаємо вибране ребро до G_{sparse} з вагою w_{ij}/qp_{ij}.
4. Повторюємо незалежно q разів з заміною та сумуванням ваги ребер, якщо ребро вибирається більше одного разу.

Даний підхід працює за час, наближений до лінійного, та отримана апроксимація графу містить $O(nlogn/q^2)$ вершин (n – кількість вершин вхідного графу).

2.1.4. Відмова від попереднього оброблення

Відмова від попереднього оброблення графу сумісної зустрічальності термів також може використовуватись в модифікованому методі кластеризації текстового корпусу.

В такому випадку замість певного наближення відбувається кластеризація цілого графу сумісної зустрічальності термів. Таким підходом гарантується стовідсоткове збереження кластерної структури, на відміну від інших підходів. Недоліками відмови від попереднього оброблення є необхідність зберігати в пам’яті весь граф, а його кластеризація буде потребувати найбільшу кількість часу, в порівнянні з іншими підходами. Проте для невеликих корпусів сумарний час роботи модифікованого методу може не погіршитись, за рахунок переходу відразу до кластеризації.

Формально, цей підхід є частковим випадком відсоткового порогу зі значенням 100 і частковим випадком глобального порогу, що дорівнює значенню відсутньої кореляції.

2.2. Кластеризація графу сумісної зустрічальності термів

Другою головною ідеєю запропонованого модифікованого методу острівної кластеризації текстів є використання іншого методу для

виконання кластеризації графу сумісної зустрічальності термів – такого, який дозволив би за необхідності виконання ексклюзивної кластеризації та/або ручне встановлення очікуваної кількості кластерів термів. Найкраще під зазначені вимоги підходить метод кластеризації графів k-medoids.

Як вже було зазначено в першому розділі, цей метод – модифікація класичного k-means методу. Звичайний k-means не може застосовуватись для задачі кластеризації графу в зв'язку з тим, що центроїдом кластеру в цьому методі може виступати будь-яка випадкова точка простору. Іншими словами, для використання k-means необхідне завдання метрики над простором спостережень (об'єктів кластеризації). В цей же час метод k-medoids накладає обмеження на центроїди кластерів – ними можуть бути лише точки наявних спостережень.

Найбільш поширений варіант реалізації k-medoids називається PAM (Partitioning Around Medoids). PAM використовує жадібний пошук, який може не знайти оптимальне рішення, проте він є швидшим, ніж повноцінний вичерпний пошук.

Алгоритм, що реалізовує PAM, виражається наступним чином:

1. На вхід алгоритм отримує граф $G = (V, E, w)$, а також число k – задану кількість кластерів.
2. Ініціалізація: обираємо випадкові k вершини в якості початкових медоїдів.
3. Для кожної точки знаходимо найближчий медоїд, формуючи початкове розбиття на кластери.
4. Знаходимо $minCost$, як значення функції втрат від початкової конфігурації.
5. Поки медоїди не стабілізуються, повторюємо наступні кроки.
6. Для кожного медоїду m повторюємо кроки 7-12.
7. Для кожної вершини $v \neq m$, що знаходиться всередині кластеру з центром в m повторюємо кроки 8-12.
8. Переміщуємо центр кластеру з m в v.

9. Перерозподіляємо всі вершини між новими медоїдами.
10. Знаходимо $cost$ – значення функції втрат від поточної конфігурації.
11. Якщо $cost < minCost$, запам'ятовуємо медоїди і прирівнюємо $minCost = cost$.
12. Повертаємо медоїд на місце (в m).
13. Робимо найкращу знайдену заміну зі всіх, що розглядались.

Таким чином, одна ітерація цього алгоритму відповідає заміні одного медоїду кластеру. За кожну ітерацію алгоритм перебирає $(n - k)$ точок графу, переміщуючи туди відповідний медоїд. Для кожної такої заміни також необхідно перерахувати відстані всіх точок до медоїдів – якщо всі попарні відстані між точками знаходяться в пам'яті, то цей етап займе $(n - k) * k$ дій. Також оптимальна заміна займе стільки же дій. Отже, складність однієї ітерації PAM в найгіршому випадку – $O(k * (n - k)^2)$. Як показали експерименти, кількість ітерацій для графа з декількома тисячами вершин знаходиться в діапазоні $[30,50]$.

Даний алгоритм може бути вдосконаленим за допомогою ще більш жадібної евристики, яка буде проводити пошук найкращої заміни лише по невеликій частині одного кластеру. Алгоритм з використанням такої евристики наступний:

1. На вхід алгоритм отримує граф $G = (V, E, w)$, а також число k – задану кількість кластерів, число s – кількість точок, які випадково обираються всередині кластеру та число $SequenceThreshold$ – поріг зупинки алгоритму.
2. Ініціалізація: обираємо випадкові k вершини в якості початкових медоїдів.
3. Для кожної точки знаходимо найближчий медоїд, формуючи початкове розбиття на кластери.

4. Знаходимо $minCost$, як значення функції втрат від початкової конфігурації.
5. Визначимо змінну, яка буде зберігати кількість ітерацій, під час якої розбиття не вдосконалювалося: $StableSequence = 0$.
6. Для кожного медоїду m робимо кроки 7-15.
7. Вибираємо випадково s точок всередині кластеру з медоїдом m.
8. Для кожної вершини v з s робимо кроки 9-13.
9. Переміщуємо медоїд з m в v.
10. Перерозподіляємо всі вершини між новими медоїдами.
11. Знаходимо $cost$ – значення функції втрат від поточної конфігурації.
12. Якщо $cost < minCost$, запам'ятовуємо медоїди і прирівнюємо $minCost = cost$.
13. Повертаємо медоїд на місце (в m).
14. Якщо найкраща заміна з s покращує функцію втрат, робимо цю заміну та приймаємо $StableSequence = 0$.
15. Інакше $StableSequence \mathrel{+}= 1$. Якщо $StableSequence > SequenceThreshold$ – зупиняємо алгоритм та повертаємо поточну конфігурацію.

Складність однієї ітерації такого алгоритму складає $O(n * k * s)$, при цьому $s \ll n/k$, що радикально зменшує обчислювальну складність алгоритму. В роботі, що розглядає подібний алгоритм, показується, що зниження параметру s до 2 і, навіть, до 1, практично не погіршує різноманітні метрики якості кластерів [21]. Це досягається тим, що кількість ітерацій такого алгоритму, в порівнянні з PAM, збільшується на порядок. Проте завдяки радикальному пришвидшенню окремої ітерації, на практиці таке збільшення кількості ітерацій не впливає на швидкодію.

Також наявні реалізації методу k-medoids, які мають повністю лінійну швидкодію і котра залежить лише від кількості вершин та

параметру k. Одна з таких реалізацій розглянута в роботі та реалізовується наступним алгоритмом [22]:

1. На вхід алгоритм отримує граф $G = (V, E, w)$, а також число k – задану кількість кластерів.
2. Для кожної вершини j обчислюємо параметр v_j за формулою (2.2):

$$v_j = \sum_{i=1}^{n} \frac{w_{ij}}{\sum_{l=1}^{n} w_{il}} \tag{2.2}$$

3. Виберемо k з найменшими значеннями v_j в якості початкових медоїдів.
4. Для кожної точки знаходимо найближчий медоїд, формуючи початкове розбиття на кластери.
5. Знаходимо $minCost$, як значення функції втрат від початкової конфігурації.
6. Знаходимо новий медоїд для кожного кластеру, який не погіршує функцію втрат.
7. Перерозподіляємо всі вершини між новини медоїдами.
8. Знаходимо значення функції втрат. Якщо вона дорівнює попередній, закінчує виконання алгоритму.

Складність ітерації цього алгоритму складає $O(n * k)$.

Також досить відомою є модифікація PAM під назвою clara. Дана модифікація в ході ітерації випадковим чином вибирає підмножину вершин і кластеризує підграф, який вони утворюють. Інші вершини просто розподіляються по найближчим медоїдам із підграфу. Втрату інформації (оскільки кластеризується лише підграф) пропонується компенсувати послідовним прогоном ітерацій на різних підмножинах вершин і вибором найкращої.

Можливо визначити безкінечну кількість способів для виділення підграфу, найпростішими з яких є:

1. повністю випадковий вибір вершин з рівномірним розподілом;
2. випадковий вибір вершин з ймовірністю, що є пропорційна степеню вершини в вхідному графі;
3. завжди вибирати фіксовану кількість вершин з найбільшим степенем, якщо не вистачило – випадково з рівномірним розподілом.

Clara показує найкращі результати на текстових корпусах з рівномірною або такою, що близька до рівномірної, структурою. В такому випадку на кожній ітерації пропонується обирати половину графу для кластеризації.

Всі розглянуті реалізації методу k-medoids вимагають явне завдання кількості кластерів, тобто в першу чергу цей метод реалізує оптимальне розбиття графу на задану кількість частин (так званий graph partitioning), а не виділення спільнот (community detection). При цьому в поставлених вимогах до модифікованого методу кластеризації текстових колекціях зазначено лише «можливість ручного визначення кількості кластерів». Таким чином, необхідно розширити метод k-medoids, додавши можливість автоматичного визначення кількості кластерів (параметру k).

Досягти це можливо двома шляхами:

1. задаючи деяку метрику «якості» розбиття і автоматизуючи процес вибору кількості кластерів, автоматично вибираючи варіант, який має найкращий результат за визначеною метрикою;
2. виводити результати кластеризації з різною кількості кластерів та делегувати вибір найкращого з представлених варіантів користувачу.

Більшість популярних метрик якості розбиття, які можливо використати в першому підході, розглянуто в розділі 4. Також можливе поєднання обох підходів – делегувати вибір користувачу найкращого

варіанту не зі всіх можливих, а з декількох, що мають найкращі результати за визначеною метрикою.

Очевидним є той факт, що повністю поняттю «автоматична кластеризація» відповідає лише перший підхід, тому на далі в модифікованому методі острівної кластеризації текстів використовується саме він.

2.3. Модифікований метод острівної кластеризації текстових колекцій

Суть методу полягає в виділенні множини термів з текстів, побудови матриці кореляції цих термів між собою (яка представляє граф сумісної зустрічальності термів), кластеризації цього графу і співставлення отриманим кластерам термів кластерів документів.

Метод кластеризації текстових колекцій складається з таких етапів:

1. Попереднє оброблення текстів з вхідної колекції документів.
2. Виділення з текстів множини термів, з яких вони складаються.
3. Побудова графу сумісної зустрічальності термів.
4. Попереднє оброблення графу сумісної зустрічальності термів і отримання його наближення.
5. Кластеризація отриманого наближення графу методом k-medoids.
6. Розбиття документів на кластери на основі отриманих кластерів термів.

Нижче описаний кожний етап даного методу детальніше.

2.3.1. Попереднє оброблення текстів з вхідної колекції документів

Нехай Q – це колекція документів, що подається на вхід методу.

На даному початковому етапі виконується видалення стоп-слів з текстів колекції Q, а також всі терми приводяться до своєї початкової форми.

Дані операції розглянуто в розділі 1. Метою їх виконання є зменшення потужності множини термів та підвищення якості подальшої кластеризації. Підвищення якості відбувається за рахунок видалення часто вживаних слів, що не мають відношення до змісту тексту, а також завдяки тому, що всі форми одного терму приводяться до однієї сутності.

2.3.2. Виділення множини термів

На цьому етапі відбувається формування множини термів T, яка буде в подальшому підлягати кластеризації.

Нехай K – це множина всіх термів, з яких складаються документи колекції Q після виконання першого етапу. Дана множина може використовуватись для подальшої кластеризації в якості множини T, проте через велику потужність цієї множини її використання може бути не доцільним. В такому випадку можливо виділити з цієї множини підмножину значущих термів K'_β.

Визначення 6. Терм t, що належить множині всіх термів K, називається значущим термом на рівні β, якщо різниця частоти входжень терму t в множину K і середньої частоти входжень цього терму в великий корпус текстів заданою мовою перевищує число β.

Таким чином, значущий терм це терм, який з великою долею ймовірності пов'язаний зі змістом документів, що складають колекцію Q. На практиці рекомендується використовувати значення β рівне 0. Множина K'_0 також може бути використана в якості множини T.

В випадку, якщо відомий результат експертної кластеризації колекції Q, або колекція Q є навчальною вибіркою для кластеризації текстів певного типу, або відомі попередні центроїди кластерів, можливе виділення з множини K'_0 частини термів для ще більшого зменшення потужності множини термів T. Такий процес описаний в роботі [18], і

базується на знаходження ваги для кожного елементу K'_0. Отримана таким чином множина K'' називається множиною ключових термів колекції Q.

2.3.3. Побудова графу сумісної зустрічальності термів

Після попередніх етапів отримано множину термів T та відома колекція документів Q. Впорядкуємо терми і документи будь-яким чином, отримавши сукупність, що наведена на (2.3).

$$\begin{cases} T = \{Term_1; Term_2; \dots; Term_{N_{terms}}\} \\ Q = \{Doc_1; Doc_2; \dots; Doc_{N_{docs}}\} \end{cases} \quad (2.3)$$

Відповідно до (2.3), розглянемо матрицю (2.4).

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1N_{docs}} \\ \vdots & \ddots & \vdots \\ a_{N_{terms}1} & \cdots & a_{N_{terms}N_{docs}} \end{pmatrix} \quad (2.4)$$

Елемент матриці a_{ij} дорівнює 1, якщо терм i належить документу j, і дорівнює 0, якщо навпаки.

Таким чином, граф сумісної зустрічальності термів задається матрицею попарних кореляцій булевих змінних a_{ij}, які відображають наявність терму i в документі j.

Ступінь кореляції між термами i та j (необхідно зауважити, що в даному випадку ці змінні не мають стосунку до індексів елементів матриці A) визначається очевидним чином: чим частіше ці терми зустрічаються разом, тим більше ці терми корельовані між собою. Визначати числову характеристику ступеню кореляції пропонується на основі поняття біноміального випадкового розподілу.

Нехай в колекції Q n – загальна кількість термів у всіх документах, що потрапили до множини T, а n_i – кількість термів з множини T в текстах, в яких зустрічається терм i. Нехай загальна кількість входжень терму j у всі тексти – N_j, а кількість входжень терму j у всі тексти, що містять терм i – N_{ij}.

Припустимо, що терми i та j не корельовано між собою та розподілені в документах незалежно один від одного. Тоді ймовірність того, що в текстах, що містять терм i, ми зустрінемо N_{ij} чи більше входжень терму j – це ймовірність отримання не менше N_{ij} успіхів в серії з N_j випробувань. При чому ймовірність успіху одного випробування дорівнює n_i/n . З закону біноміального розподілу, ця ймовірність буде виражатись формулою (2.5).

$$p_{ij} = P_B\left(N_{ij}, N_j, \frac{n_i}{n}\right) = \frac{\left(\frac{n_i}{n}\right)^{N_{ij}} \left(1 - \frac{n_i}{n}\right)^{N_j - N_{ij}} N_j!}{\left(N_j - N_{ij}\right)!} \tag{2.5}$$

Проте ймовірність p_{ij} не може бути використана напряму в якості міри кореляції термів між собою. Це пов'язано з тим, що результати підрахунку за формулою (2.5) не є симетричними, тобто $p_{ij} \neq p_{ji}$. Тому в якості міри кореляції термів між собою використовується більша з цих величин (2.6).

$$c_{ij} = \max(p_{ij}, p_{ji}) \tag{2.6}$$

Необхідність використання максимуму можливо проілюструвати наступним прикладом. Нехай в корпусі з N текстів однакового, після попередніх етапів, розміру, терм i зустрічається лише один раз в одному тексті. Терм j також зустрічається лише в цьому текстів, проте набагато частіше. Інтуїтивно зрозумілим є те, що ці терми є мало корельованими між собою. Проте в наведеному прикладі лише величина p_{ji}, яка дорівнює $1/N_{docs}$, є достатньо великою, щоб показувати цю слабку корельованість.

В результаті, граф сумісної зустрічальності термів задається квадратною, симетричною матрицею C.

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1N_{terms}} \\ \vdots & \ddots & \vdots \\ c_{N_{terms}1} & \cdots & c_{N_{terms}N_{terms}} \end{pmatrix} \quad (2.7)$$

Даний етап є найбільш обчислювально-складним етапом методу кластеризації текстових колекцій.

Описаний етап заснований на перевірці статистичної гіпотези про попарну незалежність присутності термів в документах, таким чином саме цей етап визначає статистичний характер роботи модифікованого методу кластеризації.

2.3.4. Попереднє оброблення графу сумісної зустрічальності термів і отримання його наближення

На даному етапі відбувається отримання наближення графу сумісної зустрічальності термів, який заданий отриманою на попередньому етапі матрицею C. Для цього можуть використовуватись описані в п. 2.1 підходи, а результатом виконання цього етапу є матриця C_{sparse}, що задає апроксимований граф.

Визначення 7. Першим різновидом модифікованого методу острівної кластеризації будемо називати таку реалізацію методу, яка не виконує оброблення на даному етапі, тобто $C_{sparse} = C$.

Визначення 8. Другим різновидом з параметром k модифікованого методу острівної кластеризації будемо називати таку реалізацію методу, яка на даному етапі отримує наближення графу за допомогою глобального порогу k.

Визначення 9. Третім різновидом з параметром s модифікованого методу острівної кластеризації будемо називати таку реалізацію методу, яка на даному етапі отримує наближення графу за допомогою відсоткового порогу s.

Визначення 10. Четвертим різновидом модифікованого методу острівної кластеризації будемо називати таку реалізацію методу, яка на

даному етапі для отримання апроксимації графу використовує підхід ефективних опорів до оброблення графу сумісної зустрічальності термів.

2.3.5. Отримання кластерів термів

Отримана матриця C_{sparse}, що описує апроксимацію початкового графу C, на даному етапі кластеризується за допомогою використання методу k-medoids. Як зазначено в п. 2 даного розділу, k-medoids може використовуватись в двох варіантах: з ручним завданням кількості кластерів – параметру k, та з його автоматичним знаходженням.

Визначення 11. Автоматичним різновидом модифікованого методу острівної кластеризації будемо називати таку реалізацію методу, що використовує автоматичне знаходження кількості кластерів термів.

Визначення 12. Ручним різновидом модифікованого методу острівної кластеризації будемо називати таку реалізацію методу, що використовує ручне завдання кількості кластерів.

Враховуючи визначення різновидів, які засновані на використаній процедурі попереднього оброблення графу сумісної зустрічальності термів, наприклад, перший автоматичний різновид – це така реалізація методу, яка не виконує оброблення графу та автоматично знаходить кількість кластерів.

Оскільки основним параметром, що використовується для пошуку кластерів термів, є сила зв’язку термів між собою, то і кластери термів формуються на основі їх сумісної зустрічальності. Разом в один кластер потрапляють тільки найбільш пов’язані між собою за змістом терми.

2.3.6. Співставлення кластерів документів кластерам термів

Для отримання кластерів текстів на основі отриманих кластерів термів необхідно задати критерій, який визначає приналежність кожного тексту певному кластеру термів. Для цього в описаному методі може використовуватись три вирази. В алгоритмі, що реалізує описаний метод

кластеризації, обирається з цього набору найкращий критерій для кожного конкретного практичного випадку. Далі описано ці критерії та методика вибору одного з них:

1. Відносити документ до того кластеру термів TC, у якого кількість входжень термів, що складають цей кластер, в документ, найбільша.
2. Відносити документ до того кластеру термів TC, у якого кількість входжень в документ пар термів з силою зв'язку більше заданої, що знаходяться в кластері, найбільша.
3. Відносити документ до того кластеру термів TC, у якого кількість входжень в документ трійок термів з силою зв'язку більше заданої, що знаходяться в кластері, найбільша.

Методика вибору критерію із описаних є наступною: чим більш сфокусованими за тематикою кластерів документів передбачається отримати, тим більший номер критерію необхідно вибрати. Ця методика базується на практичному сенсі вибору того чи іншого критерію. Адже, в випадку вибору першого критерію, для того, щоб віднести документ до деякого кластеру, теоретично достатньо лише одного входження одного терму. Ця умова є досить слабкою і призводить до кластеризації, в результаті якої кластери будуть мати широку тематику.

В випадку вибору другого критерію умова приналежності документу обраному кластеру посилюється, що дає більш сфокусовані за тематикою кластери текстів. Ще в більшій мірі це виконується і для третього критерію.

Висновки за другим розділом

У даному розділі розглянуто підходи до попереднього оброблення графу сумісної зустрічальності термів перед його кластеризацією, метод кластеризації графу k-medoids та його реалізації і модифікації.

Недоліки методу острівної кластеризації, розглянуті в розділі 1, виправлені шляхом пропозиції модифікованого методу острівної кластеризації текстових колекцій. Запропонований метод базується на використанні досить простої в обчислювальному сенсі процедури розбиття графу на задану кількість частин. Також розглянуто можливість автоматичного визначення необхідної кількості частин.

Запропонований модифікований метод острівної кластеризації текстових колекцій складається з таких етапів: попереднє оброблення текстів з вхідної колекції документів, виділення з текстів множини термів, з яких вони складаються, побудова графу кореляції термів між собою, попереднє оброблення графу і отримання його наближення, кластеризація отриманого наближення графу методом k-medoids, розбиття документів на кластери на основі отриманих кластерів термів. В результаті кластери документів формуються на основі наборів термів, які є сильно корельованими між собою та визначають зміст і тематику документів.

Запропоновано декілька різновидів розробленого методу, які відрізняються використаними процедурами попереднього оброблення графу сумісної зустрічальності термів та його розбиття на кластери.

Розроблений модифікований метод острівної кластеризації може використовуватися в різних галузях, де необхідна кластеризації документів на основі їх змісту, та для текстів на різну тематику. Універсальність функціонування методу забезпечується необхідністю лише мінімального набору лінгвістичних ресурсів, які використовуються лише на першому етапі попереднього оброблення текстів.

3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЧНОЇ КЛАСТЕРИЗАЦІЇ ТЕКСТОВИХ КОЛЕКЦІЙ

3.1. Основні вимоги до програмного забезпечення

Сформулюємо та опишемо основні вимоги до програмного забезпечення для автоматичної кластеризації текстових колекцій. Система, що розроблюється, повинна забезпечувати такі функціональні можливості, як:

- завантаження корпусу текстових документів для кластеризації;
- попереднє оброблення завантаженого корпусу;
- обчислення матриці кореляції термів;
- збереження обчисленої матриці кореляції термів в файл;
- завантаження попередньо обчисленої матриці кореляції термів з файлу;
- фільтрування графу сумісної зустрічальності термів обраним користувачем підходом;
- виконання кластеризації обробленого графу сумісної зустрічальності термів острівним та модифікованим методами;
- збереження отриманого результату кластеризації в текстовий файл;
- підтримка текстових документів на англійській мові.

Також, крім функціональних вимог до системи, наявні такі нефункціональні вимоги:

- система повинна працювати на комп’ютерах під управлінням ОС Windows 7 та вище;
- система повинна підтримувати можливість пакетної обробки;
- система повинна бути розширюваною, а саме підтримувати легке додавання: нових методів оброблення графу та кластеризації; інших інтерфейсів користувача;

– система повинна бути протестована за допомогою модульного та інтеграційного тестування.

Діаграма варіантів використання розробленого програмного забезпечення представлені на рис. 3.1.

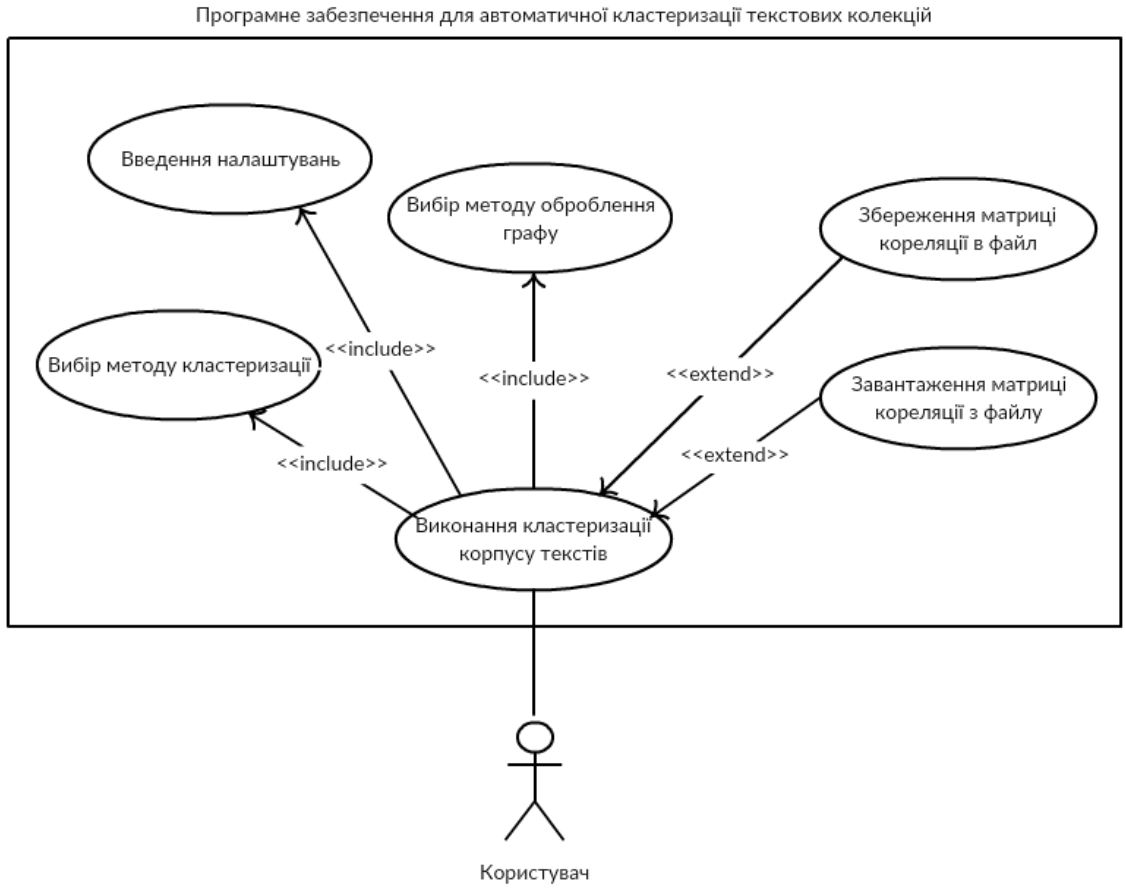

Рис. 3.1. Діаграма варіантів використання програмного забезпечення для автоматичної кластеризації текстових колекцій

3.2. Опис обраних засобів розроблення програмного забезпечення

Програмне забезпечення для автоматичної кластеризації текстових документів розроблено на платформі .NET з використанням мови програмування C#. В якості середовища розроблення використання Visual Studio 2015, а в якості системи керування версіями – git.

Коротко розглянемо обрані засоби.

3.2.1. Платформа .NET

.NET – це програмна платформа, випущена компанією Microsoft. Головними складовими даної платформи є загальномовне середовище виконання (CLR) та бібліотека класів .NET Framework (FCL).

Як зрозуміло з назви, головною задачею CLR є виконання та управління кодом під час виконання (такий код називається «керованим»), проте вона також виконує інші функції, такі як керування пам'яттю, завантаження коду, оброблення виключень, керування типізацією та безпекою [23]. Програма для .NET Framework створюється на будь-якій мові, що його підтримує, та компілюється компілятором цієї мови в проміжний байт-код Common Intermediate Language (CIL) (раніше називався Microsoft Intermediate Language, MSIL). У термінах .NET результат називається збіркою. Потім код або виповнюється віртуальною машиною CLR, або транслюється компілюється спеціальною утилітою в виконуваний код для конкретної архітектури процесора. Проте використання віртуальної машини є більш бажаним, а можливість компіляції для конкретного процесору застосовується не так часто. У разі використання віртуальної машини CLR вбудований в неї JIT-компілятор «на льоту» перетворює проміжний байт-код в машинні коди для потрібного процесора [23].

Бібліотека Framework Class Library містить об'єктні класи .NET, які є доступними для всіх підтримуваних мов програмування. Базові типи знаходяться в ядрі FCL та носять назву Base Class Library (BCL). Крім них, FCL містить класи для створення застосунків з використанням технологій Windows Forms, WPF, ASP.NET, WCF, Language Integrated Query та інші [24].

Платформа .NET має наступні переваги:

– використання компонентно-орієнтованого підходу, що значно підвищує роздільність коду та його незалежне використання;

– незалежність від мови програмування – різні частини програмного забезпечення можуть бути написаними на різних підтримуваних мовах (C#, F#, Managed C++ та інші);
– велика кількість створених спільнотою пакетів, що надають різні функціональні можливості;
– простий процес інсталяції розробленого програмного забезпечення для кінцевого користувача;
– наявність Language Integrated Query (LINQ) – мови запитів, подібної до SQL, що значно спрощує маніпуляції з даними;
– наявність бібліотеки паралельних задач (Task Parallel Library, TPL), що значно спрощує процес додавання паралелізму в програмні застосунки.

3.2.2. Мова програмування C#

C# – об’єктно-орієнтована мова програмування високого рівня, що була розроблена для використання разом з платформою .NET. На сьогодні C# є однією з найбільш поширених мов для розроблення прикладних застосунків [25].

Синтаксис даної мови є Сі-подібним і найбільш близьким до таких мов, як C++ та Java. Як будь-яка об’єктно-орієнтована мова, C# підтримує поняття поліморфізму, інкапсуляції та наслідування, а також має строгу статичну типізацію. Мова C# перейняла багато особливостей від своїх попередників (C++, Delphi, Modula і Smalltalk), проте на основі практики їх використання виключила деякі конструкції та моделі, що викликали багато проблем [26].

Дану мову обрано для розроблення програмного забезпечення в рамках даної роботи завдяки наступним перевагам:

– строга статична типізація значно зменшує кількість помилок та спрощує їх виявлення ще на етапі написання коду;
– підтримка LINQ;

– наявність таких конструкцій мови, як властивості, перевантаження операторів, атрибути;
– наявність делегатів – строго типізованих та безпечних «посилань» на функції, використання яких значно спрощу реалізацію реакцій на події;
– можливість ведення документації разом з кодом у вигляді XML коментарів, яка буде доступна всім при використанні документованого коду.

При розробленні використано передостанню версію мови – C# 6.0 – котра надає такі додаткові можливості, як null-умовні оператори, ініціалізатори автоматичних властивостей, інтерполяція рядків, імпорт статичних функцій та інші.

3.2.3. Visual Studio 2015

Microsoft Visual Studio – серія продуктів компанії Microsoft, яка де-факто є стандартом для розроблення програмних застосунків на мові C# і платформі .NET. Visual Studio – це інтегроване середовище розробки, що також включає в себе ряд інших інструментальних засобів, з можливістю їх розширення за допомогою доповнень (Add-Ins). Дана IDE дозволяє розроблювати консольні застосунки, застосунки з графічним інтерфейсом (за допомогою технологій Windows Forms та WPF), веб-сайти, веб-застосунки, веб-сервіси і інші.

Visual Studio включає в себе редактор програмного коду, що підтримує мови платформи .NET та інші, з підтримкою технології IntelliSense і можливістю рефакторингу коду. Крім цього, доступні такі інструменти, як:

– вбудований відладчик Visual Studio Debugger, що може працювати як на рівні програмного коду, так і на рівні машинних команд;

– редактор форм для розроблення застосунків з графічним інтерфейсом за допомогою технологій Windows Forms та WPF;
– веб-редактор для розроблення веб-сайтів з застосуванням технологій ASP.NET, ASP.NET MVC;
– дизайнер UML, що підтримує генерацію коду (наприклад класів), з розроблених моделей;
– дизайнер баз даних, з підтримкою підходів Model First (генерація БД з моделі) та Database First (генерація моделі з БД).

Visual Studio 2015 – це передостання на сьогодні версія Visual Studio, яка була представлена 20 червня 2015 року [27]. Ця версія містила наступні нові можливості та суттєві зміни [27]:

– підтримка .NET Framework 4.6;
– підтримка багатьох цільових платформ – крім можливості розроблення застосунків для ОС Windows була додана можливість розроблення мобільних застосунків для ОС iOS та Android (за допомогою технологій Xamarin або Apache Cordova);
– підтримка фреймворку Unity для розроблення багатоплатформних ігор;
– підтримка Universal Windows Platform – універсальної платформи Windows, що дозволяє розроблювати застосунки для будь-яких пристроїв під керуванням ОС Windows 10.

3.2.4. git

git – це розподілена система керування версіями, перша версія якої була розроблена Лінусом Торвальдсом і випущена 7 квітня 2005 року [28]. Ядро git спроектовано і розроблено у вигляді набору утиліт, які приймають налаштування за допомогою параметрів запуску, завдяки чому можливо досить просто розроблювати програми, що використовують це ядро (наприклад, графічні обгортки). Відповідні постійні налаштування (наприклад, ім’я користувача) ядро зберігає в текстових файлах [28].

В ядрі git для відстеження змін файлів використовується механізм так званих зліпків, механізм списку патчів для файлів, на відміну від Subversion та подібних систем, не використовується. Коли розробник фіксує зміни (т.з. «коммітить»), git переглядає всі файли проекту: якщо файл змінено, створюється його зліпок, інакше додається просте посилання на попередню версію. Ця база даних зліпків зберігається разом з проектом в окремій директорії з назвою «.git». git виділяє три стани файлу: змінений (проте він ще не відмічений для фіксації змін), підготовлений (відмічений для фіксації змін) та зафіксований [29].

Вся історія змін в базі даних git зберігається локально і при необхідності вивантажується у віддалений репозиторій – такий підхід характерний для будь-яких розподілених систем керування версіями. Завдяки такому підходу можлива робота програміста без постійного Інтернет-підключення, за допомогою локальної копії репозиторію [29].

3.3. Опис розробленого програмного забезпечення

В рамках даної роботи розроблено наступне програмне забезпечення:

- бібліотека оброблення природномовних текстів SimpleNetNlp;
- бібліотека-ядро автоматичної кластеризації текстових документів разом з його тестами в окремій збірці;
- консольний застосунок для автоматичної кластеризації текстових документів.

Розглянемо кожне розроблене програмне забезпечення окремо.

3.3.1. Бібліотека оброблення природномовних текстів SimpleNetNlp

Розглянуті методи кластеризації текстів, які повинні бути реалізованими розробленим програмним забезпеченням, на своєму першому кроці виконують попереднє оброблення текстів. На цьому етапі відбувається приведення слів до своєї початкової форми (так звана

лематизація), а також видалення стоп-слів. Відповідно, для виконання цієї задачі, розроблене програмне забезпечення для автоматичної кластеризації повинне використовувати деяку бібліотеку для оброблення природномовних текстів.

Однією з найбільш відомих бібліотек оброблення природномовних текстів є бібліотека Stanford CoreNLP [30], що розроблюється Стенфордським університетом [31]. Ця бібліотека надає великий та досить повний API для оброблення текстів, що включає не тільки лематизацію, а й інші функціональні можливості. В зв'язку з тим, що бібліотека Stanford CoreNLP розроблена лише для платформи Java, існує проект портування даної бібліотеки на платформу .NET – Stanford.NLP.NET [32]. Незважаючи на те, що функціональні можливості даного порту є повністю аналогічними оригіналу, ця бібліотека має наступні недоліки з точки зору розробника для платформи .NET:

– необхідність окремого завантаження та розархівування Java бібліотеки з моделями, що необхідні для виконання операцій;
– всі портовані Java пакети відображаються як простори імен типів .NET, таким чином значно засмічуючи їх;
– типи, що повертаються цією бібліотекою, також належать до портованих типів Java, тому, наприклад, додати повернутий Java тип цілого числа до типу цілого числа .NET без додаткових перетворень неможливо.

Таким чином, в рамках даної роботи постало питання розроблення власної бібліотеки оброблення природномовних текстів, що не має перелічених недоліків. Розроблена бібліотека отримала назву SimpleNetNlp і є обгорткою над розглянутою бібліотекою Stanford.NLP.NET. SimpleNetNlp є open-source бібліотекою, вихідний код якої опубліковано на сервісі GitHub (див. рис. 3.2) [33].

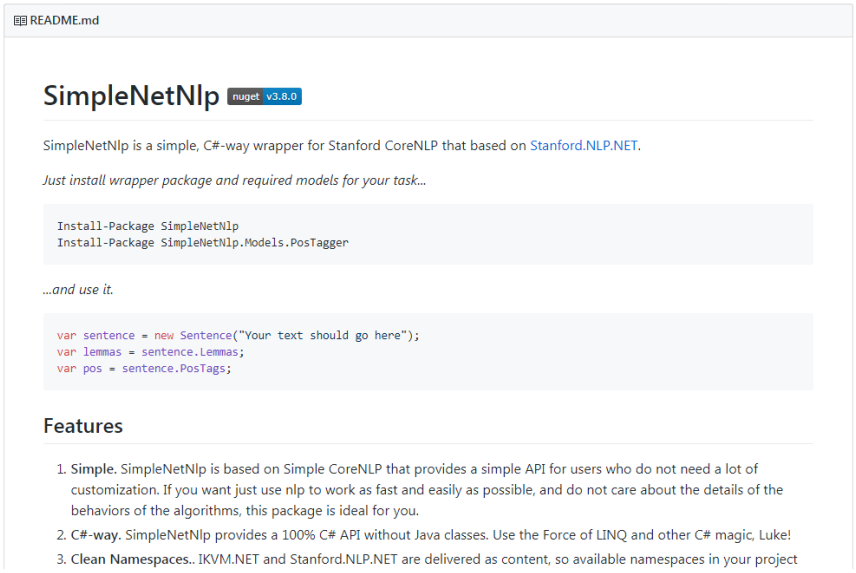

Рис. 3.2. Сторінка бібліотеки SimpleNetNlp на сервісі GitHub

Також розроблена бібліотека доступна у скомпільованому вигляді і опублікована в Nuget – системі керування версіями, що є де-факто стандартом для бібліотек .NET (див. рис. 3.3).

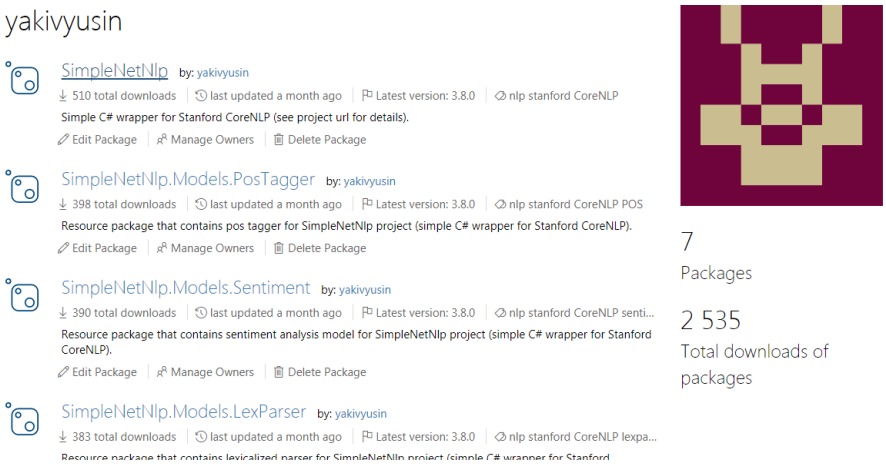

Рис. 3.3. Опублікована бібліотека в Nuget

Архітектура розробленої бібліотеки зображена на рис. 3.4. Умовно її можливо розділити на два великі окремі модулі: власне бібліотеку коду SimpleNetNlp і окремі пакети SimpleNetNlp.Models, що містять файли моделей. Таким чином, розробнику, що використовує розроблену бібліотеку, не потрібно самостійно завантажувати та надавати файли моделей для виконання операцій оброблення – дані пакети моделей роблять це самостійно за допомогою скриптів, що виконуються при побудові проекту.

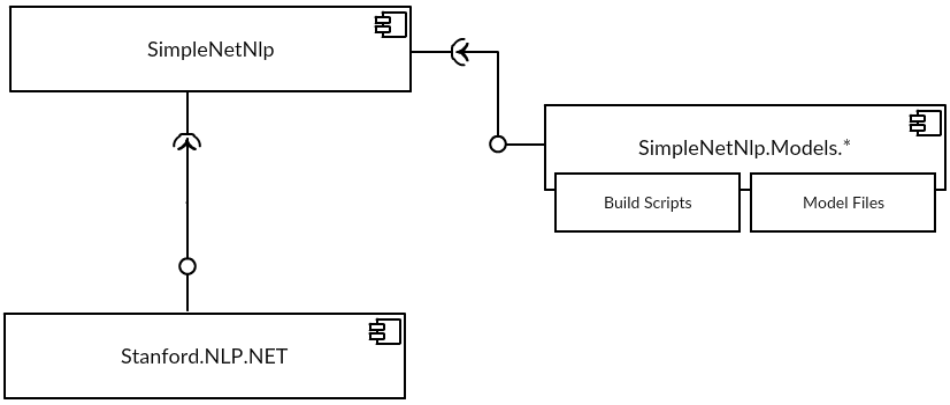

Рис. 3.4. Високорівнева архітектура бібліотеки SimpleNetNlp

Діаграма класів бібліотеки коду SimpleNetNlp наведено на рис. 3.5.

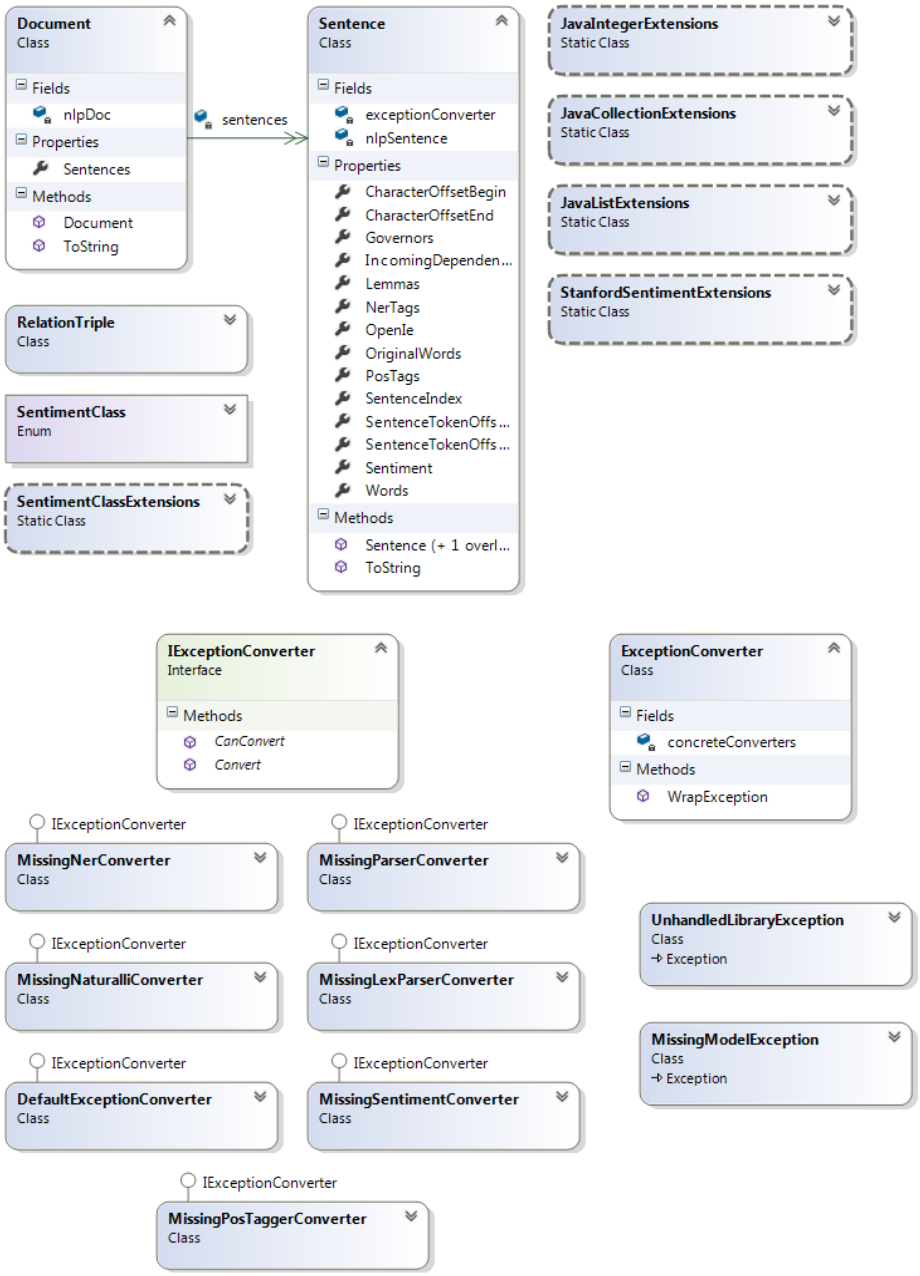

Рис. 3.5. Діаграма класів бібліотеки SimpleNetNlp

Публічно доступними для розробника, що використовує дану бібліотеку, є наступні класи:

- клас Document, який являє собою текстовий документ (екземпляри даного класу створюються, приймаючи рядок з текстом) та надає доступ до речень, з яких він складається;
- клас Sentence, який відповідно представляє речення та надає доступ до основної функціональності бібліотеки (отримання слів, лем, тегів частин речень і т.д.);
- допоміжні класи RelationTriple та SentimentClass, екземпляри яких повертаються відповідними методами класу Sentence, а також клас SentimentClassExtensions, що містить методи розширення;
- класи виключень, що визначаються бібліотекою: MissingModelException (використовується при відсутності певних файлів моделей) та UnhandledLibraryException (виключення-обгортка для непередбачених Java-виключень).

Інші зображені класи є внутрішніми класами бібліотеки, що використовуються для:

- конвертування виключень, що генеруються бібліотекою Stanford.NLP.NET, у власні виключення. Відповідальним за це є клас ExceptionConverter, котрий зберігає посилання на всі класи, що реалізовують інтерфейс IExceptionConverter, і конвертує виключення за їх допомогою. Якщо виключення є невідомим (не конвертується жодним з конверторів виключень моделей), його конвертує DefaultExceptionConverter;
- конвертування Java-типів в типи .NET. Відповідальними за це є статичні класи з назвою, що закінчується на «Extensions», і які містять методи розширення Java-типів.

3.3.2. Бібліотека-ядро автоматичної кластеризації текстових документів ClusteringCore

Відповідно до поставлених вимог до розробленого програмного забезпечення (особливо в частині розширюваності), вся логіка кластеризації текстових документів інкапсульована в окремій бібліотеці ClusteringCore.

Частина діаграми класів, що зображує публічно доступні класи, наведено на рис. 3.6.

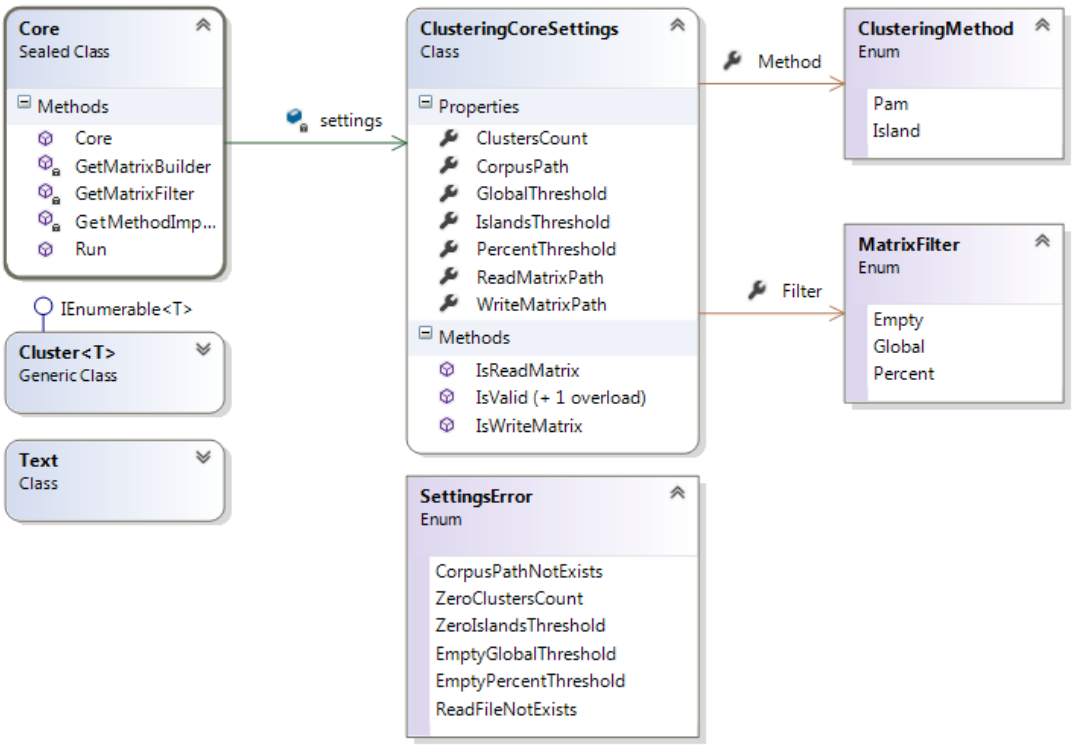

Рис. 3.6. Публічна частина бібліотеки ClusteringCore

Точкою входу для використання бібліотеки є клас Core, який виступає фасадом для всієї бібліотеки і приймає в якості параметра конструктора налаштування. Ці налаштування містять такі дані, як шлях до колекції текстових документів, обраний метод кластеризації та оброблення графу, параметри для їх роботи, шлях до збереженої матриці кореляції (якщо потрібно) та шлях для збереження матриці (якщо потрібно). Єдиний публічно доступний метод «Run» виконує кластеризацію відповідно до отриманих налаштувань (створюючи

відповідні екземпляри класів за допомогою фабричних методів) та повертає отримані кластери документів.

Реалізацію доступних методів кластеризації текстових документів (метод острівної кластеризації та модифікований метод) в бібліотеці організовано відповідно до шаблону проектування «Шаблонний метод». Це дозволило один раз визначити спільні кроки обох методів, реалізовуючи лише відмінну частину. Відповідна діаграма класів наведена на рис. 3.7.

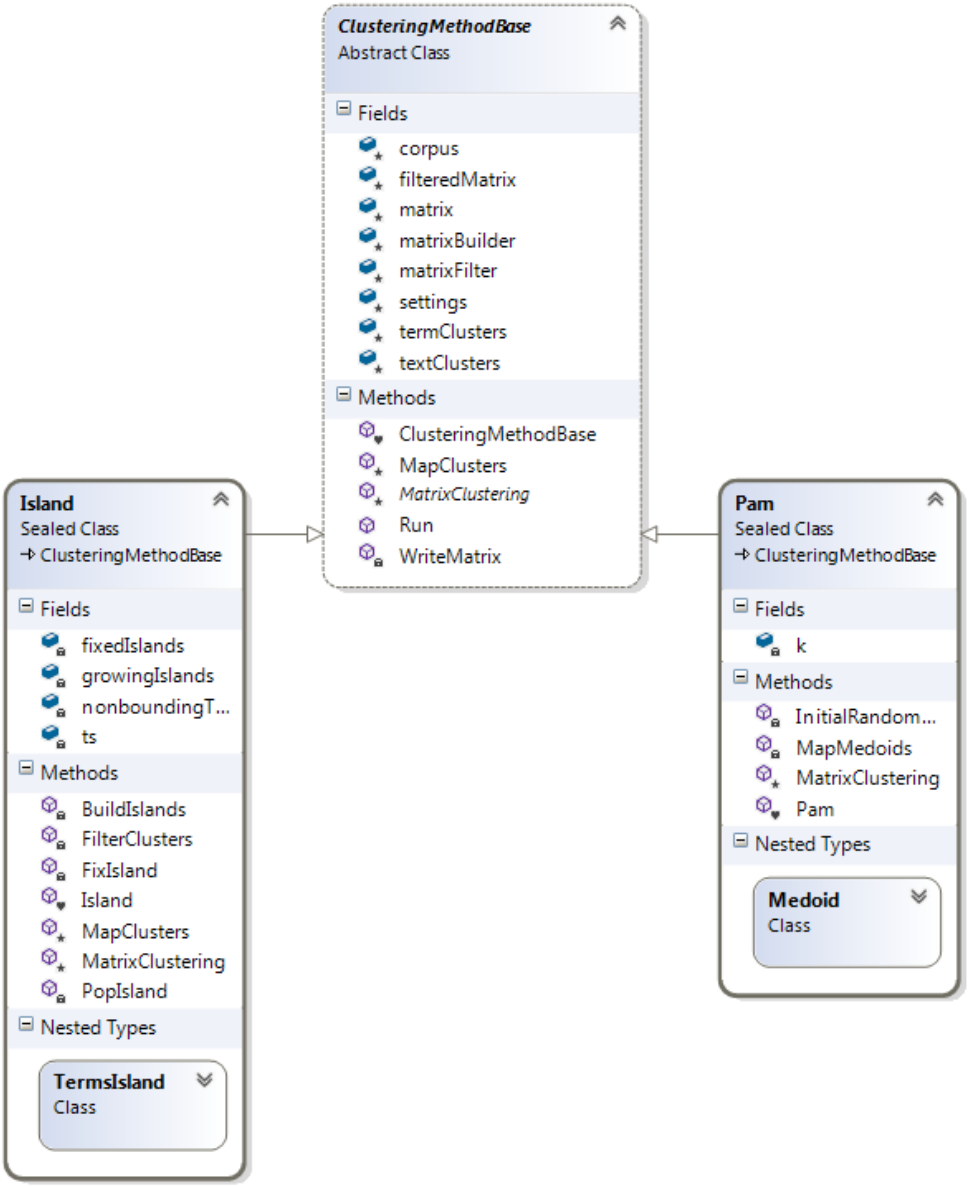

Рис. 3.7. Організація методів кластеризації

Методи оброблення графу сумісної зустрічальності термів організовані у ієрархію зі спільним предком у вигляді абстрактного класу.

Така їх організація дозволяє використовувати шаблон проектування «Стратегія» для передавання екземплярів до класів, що реалізовують методи кластеризації текстових документів. Відповідна діаграма класів наведена на рис. 3.8.

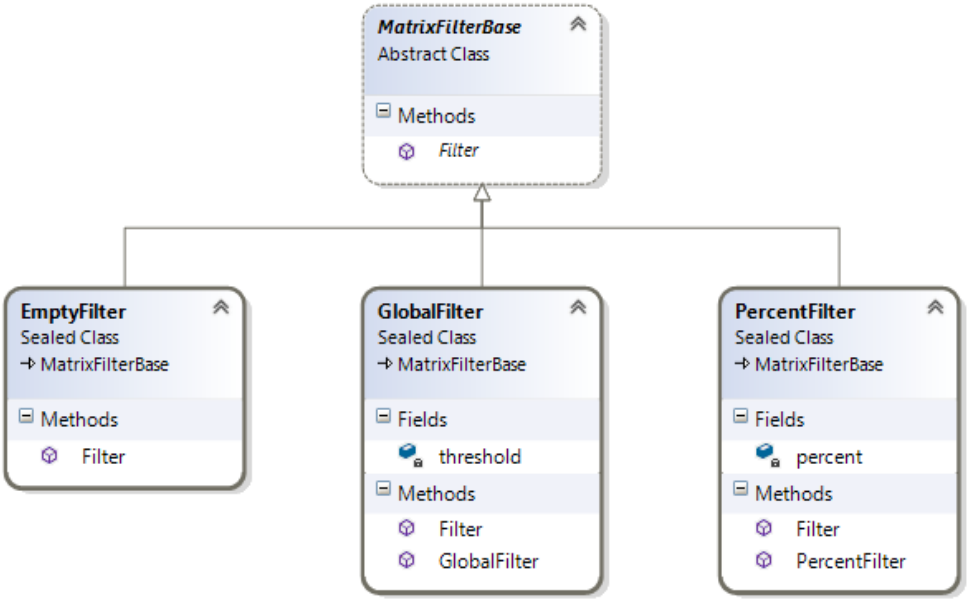

Рис. 3.8. Організація методів оброблення графу сумісної зустрічальності термів

Сам граф сумісної зустрічальності термів подається у вигляді матриці кореляції термів, що його задає, і яка зображена на рис. 3.9.

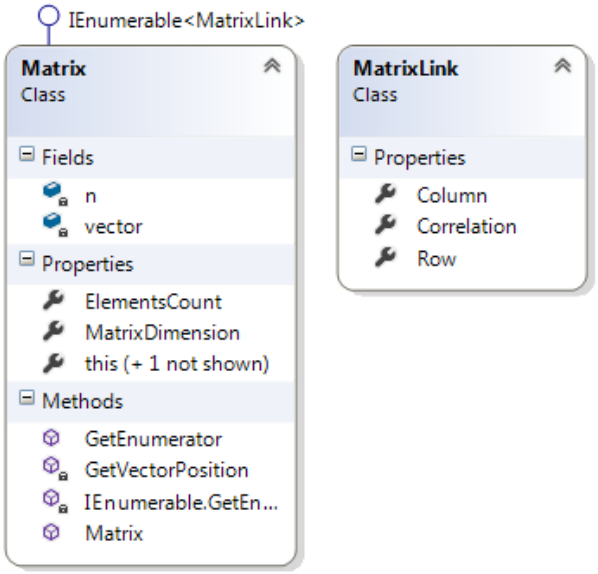

Рис. 3.9. Подання графу сумісної зустрічальності термів в бібліотеці

Оскільки ця матриця є квадратною симетричною матрицею, а значення елементів на діагоналі не є важливими (не використовуються жодними методами кластеризації), то матриця всередині класу зберігається у вигляді одномірного масиву, що містить елементи над головною діагоналлю матриці. Доступ до цих елементів організований за допомогою індексаторів – використовуючи як прямий індекс в масиві (в деяких класах для прискорення обчислень), так і двомірний індекс елементу в матриці кореляції термів (в такому випадку відбувається перетворення двомірних координат в індекс елементу в масиві).

Також клас матриці реалізовує інтерфейс IEnumerable, надаючи нумератор для перебору екземплярів спеціального класу MatrixLink, що представляють елементи матриці в об’єктно-орієнтованому вигляді.

Відповідно до вимог програмне забезпечення може або обчислити матрицю кореляцію термів на основі текстової колекції, або зчитати обчислену раніше матрицю з файлу. Тому ці варіанти отримання матриці кореляції термів організовані аналогічно до методів оброблення графу сумісної зустрічальності термів – в ієрархію зі спільним предком (див. рис. 3.10). Таким чином, їх використання також відбувається в рамках шаблону проектування «Стратегія».

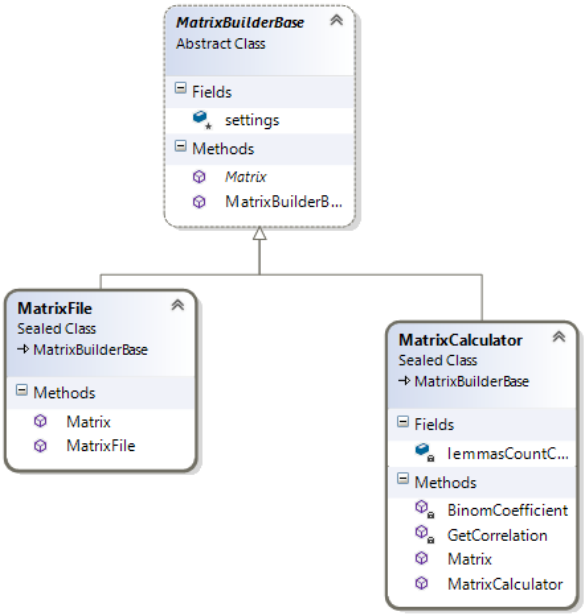

Рис. 3.10. Ієрархія методів отримання матриці кореляції термів

Обчислення матриці кореляції термів відбувається за допомогою розпаралелювання частини алгоритму та з використанням кешу частини проміжних значень, що можуть бути використані повторні. Ці способи дозволяють пришвидшити обчислення матриці кореляції термів приблизно у 2.2 рази, в порівнянні з оригінальним алгоритмом обчислення [34].

Сама колекція текстових документів також подається в об’єктно-орієнтованому вигляді за допомогою класу, для створення екземплярів якого передається шлях до папки з текстами. Абстрактний предок цього класу оголошено для зручності виконання модульного тестування (заміни текстової колекції заглушкою). При створенні екземплярів текстів, відбувається отримання вмісту файлу та його попереднє оброблення – отримання лем, видалення стоп-слів (їх перелік див. додаток 1) та знаків пунктуації. Дана структура класів, що відповідає за відображення текстової колекції, наведена на рис. 3.11.

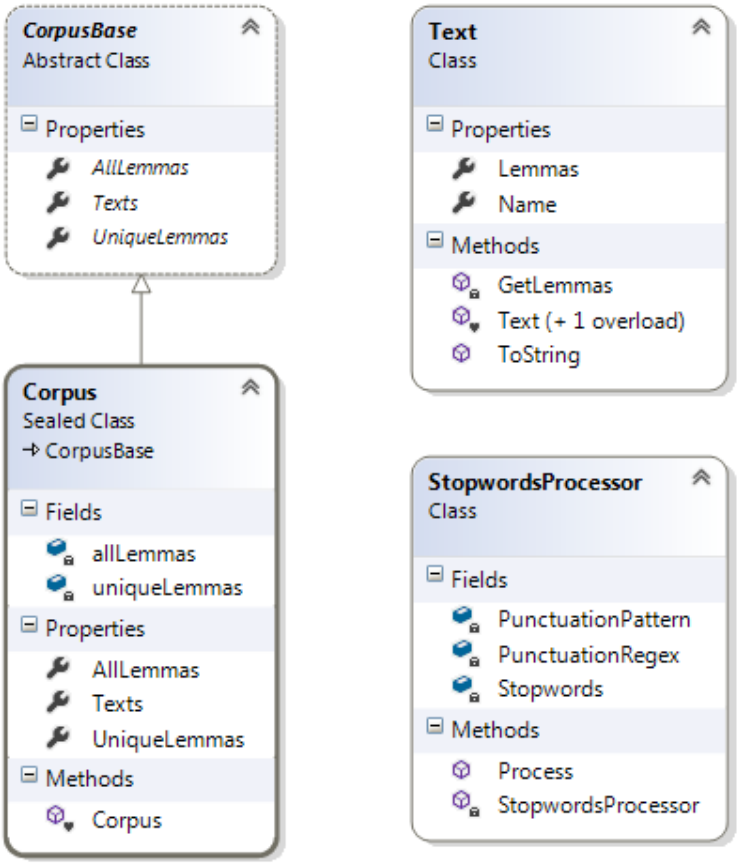

Рис. 3.11. Представлення текстової колекції

Отже, наведемо приклад використання розробленої бібліотеки для автоматичної кластеризації текстів ClusteringCore – див. лістинг 3.1.

Лістинг 3.1

Приклад використання розробленої бібліотеки

```
using ClusteringCore.Model;
using ClusteringCore;

var settings = new ClusteringCoreSettings
{
      // Зазначаємо деякі коректні налаштування
};
var core = new Core(settings);
var result = core.Run();
```

Модульні тести бібліотеки створені за допомогою MSTest (бібліотека для тестування, що встановлюється за замовчуванням разом з Visual Studio) та виділені в окремий проект ClusteringCore.Tests. Також використано бібліотеку Moq для створення заглушок об’єктів.

Створені модульні тести наведено на рис. 3.12 та містять тестування наступної функціональності:

– конвертування двомірних координат матриці в індекси в масиві;
– оброблення графу сумісної зустрічальності термів різними підходами;
– обчислення матриці кореляції термів;
– зчитування матриці кореляції термів з файлу.

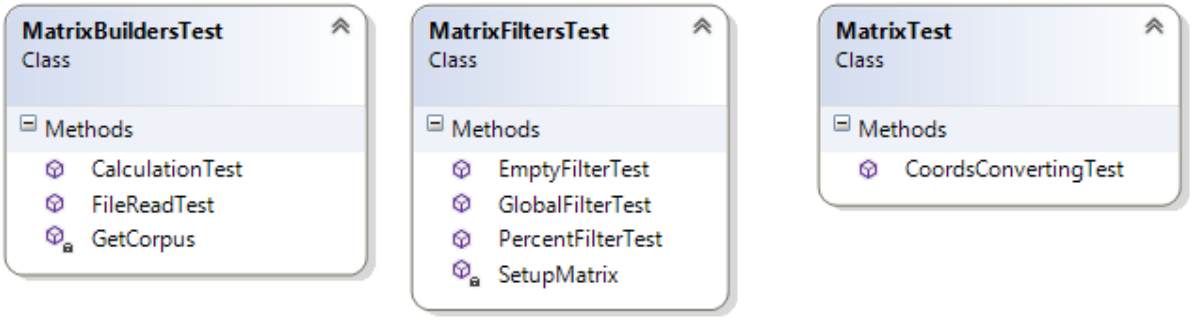

Рис. 3.12. Структура модульних тестів бібліотеки

3.3.3. Консольний застосунок для автоматичної кластеризації текстових документів

В якості кінцевого застосунку для автоматичної кластеризації текстових документів розроблено застосунок з інтерфейсом командного рядка. Такий варіант реалізації користувацького інтерфейсу дозволяє легко підтримувати пакетну обробку текстових колекцій, а також повну автоматизацію виконання кластеризації. Наприклад, консольний застосунок легко викликати з різних автоматичних скриптів, передаючи різні налаштування, що значно спростило подальше тестування розроблених методів та підходів.

Сама структура консольного застосунку є досить простою (див. рис. 3.13), оскільки вся функціональність, що стосується кластеризації, виділена в окрему бібліотеку. Таким чином, відповідальністю консольного застосунку є лише зчитування налаштувань та вивід результатів.

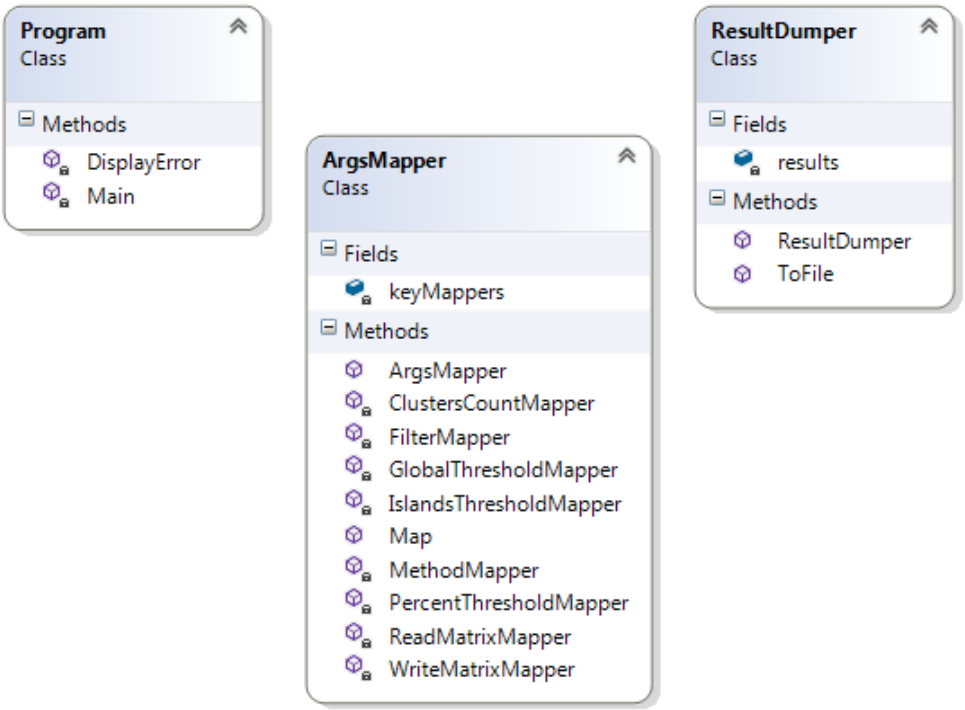

Рис. 3.13. Структура консольного застосунку

Консольний застосунок приймає налаштування кластеризації у вигляді параметрів командного рядка при запуску застосунку. Відповідно,

клас ArgsMapper приймає масив параметрів у вигляді рядків та виконує їх перетворення у об’єкт класу бібліотеки ClusteringCore ClusteringCoreSettings (який надає публічний метод IsValid() для валідації). В випадку, якщо після перетворення отримано не коректні налаштування бібліотеки кластеризації, консольний застосунок переходить до показу помилок і завершує свою роботу.

Доступні параметри командного рядку та їх допустимі значення наведено в табл. 3.1.

Таблиця 3.1

Доступні параметри запуску консольного застосунку

№ п/п	Параметр	Тип параметру	Призначення
1	Без ключа, перший	Шлях до існуючої папки	Шлях до папки, що містить колекцію текстових документів
2	-m	Рядок «pam» або «island»	Вказує на використовуваний метод кластеризації – модифікований чи оригінальний
3	-k	Додатне ціле число	Кількість кластерів, використовується при виборі модифікованого методу кластеризації
4	-ts	Додатне ціле число	Поріг росту островів, використовується при виборі оригінального методу кластеризації
5	-f	Рядок «empty», «global» або «percent»	Вказує на використовуваний підхід до оброблення графу сумісної зустрічальності термів
6	-gt	Дійсне число	Значення порогу, що використовується при виборі глобального підходу до оброблення графу
7	-pt	Додатне ціле число	Значення порогу, що використовується при виборі відсоткового підходу до оброблення графу
8	-rm	Шлях до існуючого файлу	Файл для зчитування попередньо обчисленої матриці кореляції термів

Продовження табл. 3.1

9	-wm	Шлях до файлу	Файл для збереження обчисленої матриці кореляції термів

При коректних налаштуваннях застосунок передає їх до бібліотеки кластеризації, яка виконує кластеризацію. Отримані результати передаються до екземпляру класу ResultDumper і викликається обраний метод збереження результату. В рамках даної роботи розроблений консольний застосунок підтримує збереження результатів тільки до текстового файлу відповідного формату (метод ToFile()). Проте в майбутньому легко розширити перелік підтримуваних форматів (наприклад, збереження до бази даних), а користувач буде обирати потрібний за допомогою параметру командного рядка.

Приклад текстового файлу з результатами наведено в лістингу 3.2.

Лістинг 3.2

Приклад файлу результатів

```
Cluster #1
-corpus\1.txt
-corpus\2.txt
-corpus\3.txt

Cluster #2
-corpus\4.txt
-corpus\5.txt

…

Corpus #N
…
```

Деякі приклади коректних налаштувань з поясненнями наведено в табл. 3.2.

Таблиця 3.2

Приклади параметрів запуску консольного застосунку

№ п/п	Параметри запуску	Опис виконуваних дій
1	./corpus -m pam -k 2 -f empty -wm 1.saved	Кластеризація модифікованим методом з кількістю кластерів 2, відсутнім обробленням графу та записом обчисленої матриці кореляції до файлу
2	./corpus –m island -ts 9 -f global -gt 0.001 -rm 1.saved	Кластеризація оригінальним методом з параметром 9, глобальним обробленням графу (поріг 0.001) і завантаження матриці кореляції з файлу
3	./corpus -k 2	Те саме, що пункт 1, тільки обчислена матриця кореляції термів не зберігається до файлу

Висновки за третім розділом

У даному розділі розглянуто розроблене програмне забезпечення, яке призначене для автоматичної кластеризації текстових колекцій за допомогою оригінального методу острівної кластеризації та модифікованого методу (запропонованого в даній роботі).

Можливо зробити наступні висновки про розроблене програмне забезпечення в цілому:

– розроблене програмне забезпечення можливо використовувати для автоматичної кластеризації текстових документів;
– розроблений консольний застосунок надає можливості налаштування параметрів кластеризації в залежності від потреб користувача;
– розроблене програмне забезпечення в повній мірі підтримує можливості пакетної обробки та виконання кластеризації в автоматичному режимі, завдяки реалізації у вигляді консольного застосунку;

– архітектура розробленого програмного забезпечення дозволяє легко додавати реалізації інших методів автоматичної кластеризації та підходів до оброблення графу сумісної зустрічальності термів, завдяки використанню таких шаблонів проектування, як «Шаблонний метод» та «Стратегія»;
– завдяки виділенню всього, що стосується саме кластеризації, в окрему бібліотеку, розроблене програмне забезпечення не залежить від інтерфейсу користувача і може використовуватись з будь-яким інтерфейсом (консольним, графічним, веб і т.д.) або як складова частина окремого більшого програмного комплексу.

4. АНАЛІЗ ЕФЕКТИВНОСТІ МОДИФІКОВАНОГО МЕТОДУ ОСТРІВНОЇ КЛАСТЕРИЗАЦІЇ ПРИРОДНОМОВНИХ ТЕКСТОВИХ ДАНИХ

4.1. Основні способи оцінювання методів кластеризації

Оскільки зазвичай результати кластеризації текстів інтерпретуються безпосередньо людиною, то вона повинна розуміти зміст знайденого кластеру і чому певні тексти були віднесені саме до нього – саме це є фактором, що відрізняє задачу кластеризації текстів від задач кластеризації інших видів даних і ускладнює оцінювання якості отриманих результатів кластеризації. Деякі автори зазначають, що задача оцінювання (або також вживаним є термін «валідація») якості отриманих результатів кластеризації є такою же складною, як власне кластеризація [35].

Основні способи оцінювання якості отриманих результатів кластеризації поділяються на наступні чотири види [36]:

- внутрішні оцінки;
- зовнішні оцінки;
- ручні оцінки;
- опосередкована оцінка шляхом оцінювання корисності виконання кластеризації на практиці.

Коли результат кластеризації оцінюється на основі самих отриманих кластерів, то такий вид оцінювання називається внутрішніми оцінками. Ці методи зазвичай присвоюють найкращий результат методу, який створює кластери з високою подібністю текстів в кластері і низькою подібністю між кластерами. Одним із недоліків використання внутрішніх оцінок якості отриманих результатів кластеризації є те, що високі оцінки за ними не обов’язкового призводять до ефективного пошуку інформації [37]. Крім того, формула внутрішньої оцінки сама по собі може виступати в якості критерію для розбиття корпусу на кластери – можливо створити штучний метод, який буде показувати на такій оцінці найкращі можливі результати.

Саме тому внутрішні оцінки найкраще підходять для того, щоб отримати уявлення про ситуації, коли один метод працює краще другого, але це не буде означати, що цей метод дає більш достовірні результати [38].

Найбільш популярними внутрішніми оцінками якості отриманих результатів кластеризації є індекс Девіса-Болдуіна [39], індекс Данна [40] та індекс оцінки силуету [41].

Індекс Девіса-Болдуіна обчислюється за формулою (4.1), де n – кількість кластерів, c_x – центроїд кластеру x, σ_x – середня відстань всіх елементів кластеру x до центроїду c_x, а $d(c_i, c_j)$ це відстань між центроїдами c_i та c_j.

$$DB = \frac{1}{n}\sum_{i=1}^{n} \max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \tag{4.1}$$

Чим менше значення цього індексу, тим кращим прийнято вважати отриманий результат кластеризації.

Індекс Данна обчислюється за формулою (4.2), де n – кількість кластерів, $d(i,j)$ це відстань між двома кластерами, а $d`(k)$ це відстань всередині кластеру. $d(i,j)$ може бути виміряна різними способами, наприклад в якості неї може бути прийнята відстань між центроїдами відповідних кластерів. Аналогічно, в якості $d`(k)$ також можуть бути використані різні показники, наприклад, максимальна відстань між парою точок всередині кластеру.

$$D = \frac{\min\limits_{1 \leq i < j \leq n} d(i,j)}{\max\limits_{1 \leq k \leq n} d`(k)} \tag{4.2}$$

Отриманий результат кластеризації є тим кращим за цим критерієм, чим вище отримане значення за формулою.

Індекс оцінки силуету це середній показник значень «силуету» всіх елементів корпусу. «Силует» кожного елементу визначається наступним чином: нехай елемент x_j належить кластеру c_p. Тоді якщо ми позначимо

середню відстань від цього елементу до інших елементів цього ж кластеру як a_{xj}, а середню відстань від елементу x_j до елементів іншого кластеру c_q як d_{qj}, то «силует» елементу x_j буде обчислюватись за формулою (4.3), де $b_{xj} = \min_{x \neq q} d_{qj}$.

$$S = \frac{b_{xj} - a_{xj}}{\max(a_{xj}, b_{xj})} \tag{4.3}$$

Кращі результати кластеризації характеризуються максимальним значенням індексу оцінки силуету. Також на практиці використовуються варіації цієї оцінки: спрощений силует і альтернативний силует [42].

При використанні зовнішніх оцінок результати кластеризації оцінюються на основі даних, які не використовувались для кластеризації, наприклад, відомих міток класів чи зовнішніх контрольних показників. Такі контрольні показники складаються з набору попередньо кластеризованих текстових колекцій, і ці набори часто створюються вручну людьми-експертами [35]. Цей вид оцінок результатів кластеризації визначають, наскільки близько отриманий результат відноситься до попередньо визначених класів тестів. Звісно, можлива ситуація, коли реальні результати використання методу кластеризації на практиці будуть досить далекими від отриманих показників на тестових даних. Крім цього, з точки зору пошуку інформації (використання кластеризації для аналізу текстового корпусу) відтворення відомих знань може не обов’язково бути бажаним результатом.

Ряд зовнішніх оцінок адаптований із варіантів, що використовуються для оцінювання результатів виконання класифікації. Замість підрахунку кількості разів, коли точці даних було правильно визначено клас (відомі як істинно-позитивні результати), такі оцінки підраховують кількість пар точок даних (в нашому випадку текстових документів), які було правильно віднесено до одного кластеру. Аналогічно адаптується підрахунок істинно-негативних, хибно-негативних та хибно-позитивних результатів.

До найбільших поширених зовнішніх оцінок слід віднести чистоту, міру Ренда, точність та повноту (і похідні від них оцінки, такі як F-міра), індекс Жаккара та інші.

Чистота – це міра ступеню, в якому кластери містять лише один клас [37]. Розрахунок цього показника можливо розглянути наступним чином: для кожного кластеру підраховується кількість точок даних (текстових документів) із найбільш поширеного класу у вказаному кластері. Після цього підрахуємо суму по всіх кластерам і розділимо на загальну кількість документів. Формально це можливо виразити формулою (4.4), де M це множина отриманих кластерів, D це множина класів тестового корпусу, а n це кількість документів в корпусі.

$$purity = \frac{1}{n}\sum_{m\in M}\max_{d\in D}|m \cap d| \tag{4.4}$$

Індекс Ренда обчислює, наскільки отримані кластери близькі до еталонного результату кластеризації. Можливо також розглядати індекс Ренда як відсоток правильних рішень, що прийняті методом кластеризації. Індекс Ренда обчислюється за формулою (4.5) [43].

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.5}$$

Точність системи в границях класу – це частка документів, які дійсно належать до цього класу відносно всіх документів, які було віднесено до цього класу. Повнота системи – це частка знайдених документів, що належать класу відносно всіх документів цього класу в тестовій вибірці. Таким чином точність P та повнота R обчислюється за формулою (4.6).

$$P = \frac{TP}{TP + FP};\ R = \frac{TP}{TP + FN} \tag{4.6}$$

На основі точності та повноти будуються похідні від них оцінки, найбільш популярною з яких є F-міра, яка обчислюється за формулою (4.7).

$$F_\beta = \frac{(\beta^2 + 1) * P * R}{\beta^2 * P + R} \quad (4.7)$$

Відповідно до формули, коли $\beta = 0$ F-міра дорівнює точності. Зі збільшенням параметру β збільшується вплив повноти системи. Деякі значення β отримали власні назви – наприклад F-міра з $\beta = 1$ отримала назву «індекс Соренсена».

Індекс Жаккара використовується для визначення подібності двох наборів даних (еталонного результату кластеризації та отриманого результату досліджуваним методом) і приймає значення між 0 (набори даних не мають спільних елементів) та 1 (набори даних повністю подібні). Індекс Жаккара обчислюється за формулою (4.8).

$$J(A, B) = \frac{TP}{TP + FP + FN} \quad (4.8)$$

Інколи, коли еталонний результат має специфічні особливості, обчислення вище згаданих показників можливо спростити. Наприклад, коли еталонний текстовий корпус має кластери однакового розміру, які не пересікаються, найбільш простою оцінкою якості отриманого результату кластеризації буде відсоток текстових документів, що були вірно розподілені по своїм кластерам.

Ручне оцінювання якості отриманого результату кластеризації полягає в оцінці отриманих кластерів людиною-експертом. І хоча така оцінка може бути досить інформативною та якісною, особливо при визначенні поганих кластерів, її недоліком є суб’єктивність та досить великі витрати часу.

Також одним із важливих показників, що використовується при оцінці методів кластеризації, але який не належить до оцінювання якості отриманих результатів, є швидкість виконання кластеризації. Часто цей показник використовують в парі з деякими з оцінок якості отриманих результатів для вибору певного методу кластеризації. Наприклад, якщо деякий метод в порівнянні з іншим виконує на 2% більш якісну

кластеризацію, проте потребує для цього в два рази більше часу, обраним для застосування на практиці може бути менш якісний метод.

Аналогічно до зовнішніх оцінок, швидкість виконання кластеризації вимірюється на деяких тестових текстових колекціях.

4.2. Аналіз результатів оцінювання запропонованого модифікованого методу острівної кластеризації

Оцінювання ефективності запропонованого модифікованого методу острівної кластеризації за критеріями якості отриманих результатів та швидкодії відбувалося за допомогою розробленого в рамках даної роботи програмного забезпечення для автоматичної кластеризації текстових колекцій. Перевірка відбувалася на двох корпусах документів.

Перший корпус ***B*** складається з 50 текстів, присвячених Євробаченню та діяльності компанії SpaceX, відібраних з сайту BBC [44]. В даному корпусі текстів обох тематик порівну – по 25 новин.

Другий корпус ***R*** складається з 574 текстів, розподілених порівну між сімома різними тематиками. Тексти цього корпусу є попередньо обробленою підмножиною популярного тестового набору Reuters-21578 [45]. Попереднє оброблення даної підмножини полягало в приведені текстів до формату, з яким працювала програмна реалізація (виділення міток кластерів, перейменування файлів з текстами).

У зв’язку з простотою обох тестових корпусів (кількість текстів кожного кластеру є однаковою, кожен текст належить лише до одного кластеру) в якості міри якості результатів кластеризації використано просте відношення кількості текстів, розподілених правильно по кластерам, до загальної кількості текстів в корпусі.

В якості еталону для оцінювання запропонованого модифікованого методу кластеризації текстів обрано оригінальний метод острівної кластеризації. Оцінювання відбувалося в три етапи:

1. спочатку проведено тестування запропонованих підходів до попереднього оброблення графу сумісної зустрічальності термів;
2. наступним проведено оцінювання використання методу k-medoids для кластеризації графу сумісної зустрічальності термів;
3. на останньому етапі відбулося оцінювання запропонованого модифікованого методу острівної кластеризації текстів в цілому.

Такий підхід до оцінювання дозволив окремо оцінити вклад кожної гіпотези з модифікації методу острівної кластеризації текстів, що використовуються запропонованим модифікованим методом острівної кластеризації текстів.

Запропоновані підходи до попереднього оброблення графу сумісної зустрічальності термів протестовано з наступними налаштуваннями:

– підхід з використанням глобального порогу – поріг обчислено за формулою (2.1), таким чином це відповідає оригінальному підходу до попереднього оброблення графу;
– підхід з використанням відсоткового порогу – обрано значення 15 як таке, що вдвічі перевищує знайдену верхню межу, якій відповідає глобальний поріг (див. п. 2.1.2).

Якість отриманих результатів кластеризації з використанням запропонованих підходів до попереднього оброблення графу сумісної зустрічальності термів наведена на рис. 4.1. Як бачимо, найгіршу якість отриманий результат має при використанні оригінального підходу до оброблення графу, найкращу – при відмові від попереднього оброблення. Отримані результати повністю відповідають теоретичним припущенням, описаним в п. 2.1.

Результати оцінювання швидкості виконання базової острівної кластеризації тестових корпусів з використанням запропонованих підходів до попереднього оброблення графу сумісної зустрічальності термів

наведено на рис. 4.2. Значення на рисунку подані у вигляді співвідношення часу виконання кластеризації з використанням певного підходу до часу виконання кластеризації з використанням оригінальної процедури. Таким чином, чим ближче до 1 значення, тим показана краща швидкість кластеризації. Як показало тестування, найвищу швидкість має використання відсоткового підходу – він лише на 7-9% повільніший, ніж оригінальний. Найгіршу швидкість, як очікувалось, отримано при відмові від попереднього оброблення графу сумісної зустрічальності термів, проте в випадку досить малого тестового корпусу різниця є досить малою.

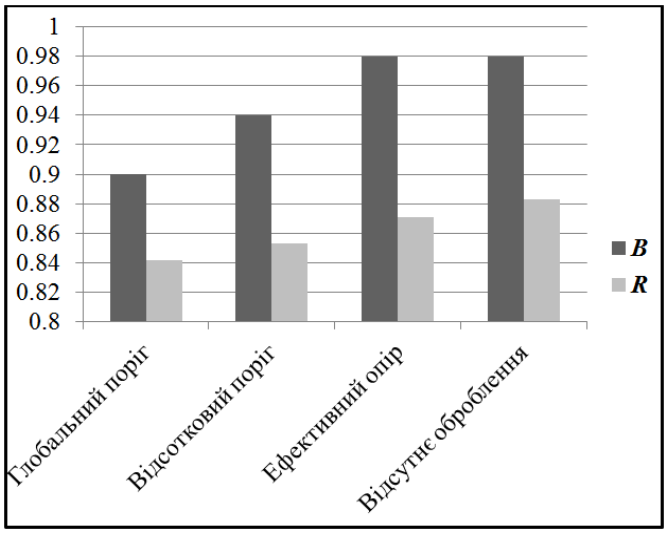

Рис. 4.1. Якість отриманих результатів кластеризації з використанням запропонованих підходів до оброблення графу

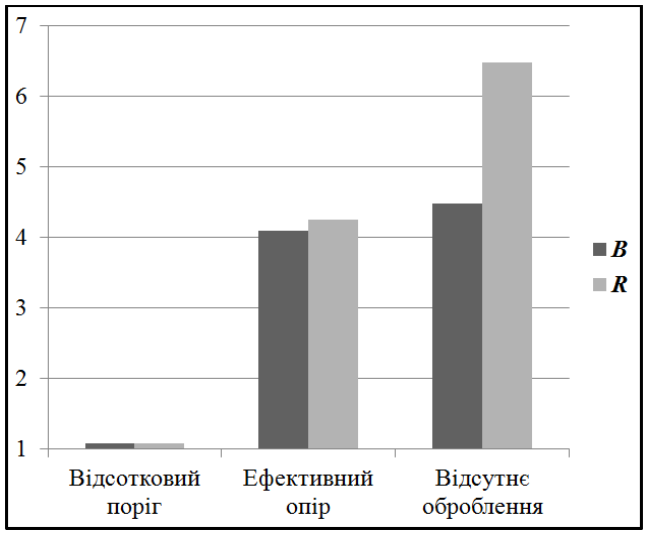

Рис. 4.2. Швидкість виконання острівної кластеризації з використанням запропонованих підходів до оброблення графу

Результати оцінювання якості отриманих результатів кластеризації з використанням методу k-medoids для кластеризації отриманого наближення графу сумісної зустрічальності термів наведено на рис. 4.3. В ході тестування проаналізовано три різні реалізації цього методу:

– PAM;

– жадібна s-евристика;

– clara.

Як очікувалося, використання методу k-medoids для кластеризації отриманого наближення графу сумісної зустрічальності термів забезпечило кращий результат кластеризації для всіх реалізацій, ніж оригінальний метод острівної кластеризації, завдяки ручному встановленню очікуваної кількості кластерів.

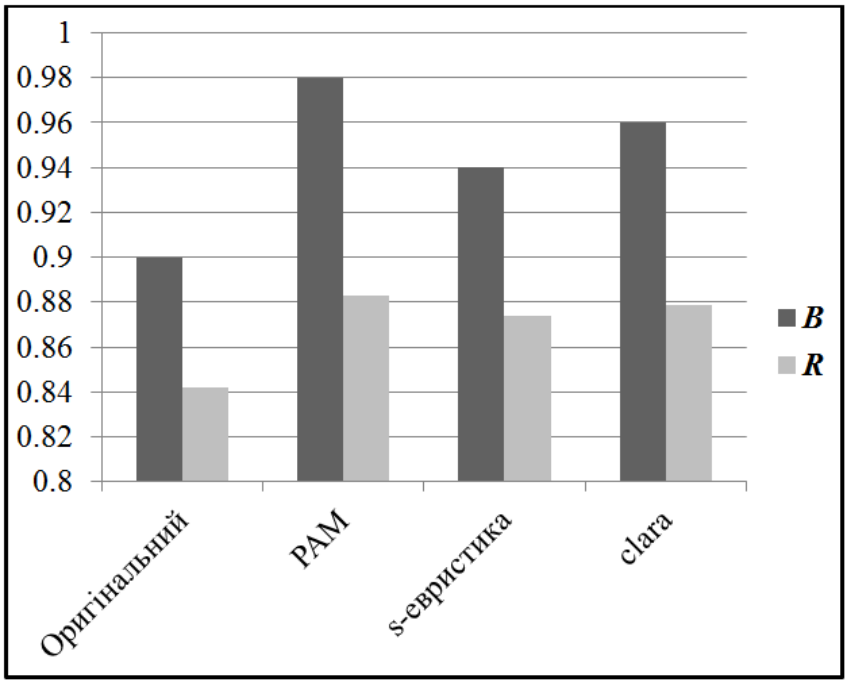

Рис. 4.3. Якість отриманих результатів кластеризації з використанням методу k-medoids для кластеризації графу

Результати тестування швидкості виконання кластеризації тестових корпусів з використанням методу, що використовує k-medoids, наведені на рис. 4.4. Всі значення подані у вигляді співвідношення часу виконання кластеризації k-medoids методом до часу виконання кластеризації оригінальним острівним методом (в %). Таким чином, чим ближче до 0

значення, тим показана краща швидкість кластеризації (менше уповільнення, в порівнянні з оригінальним острівним методом).

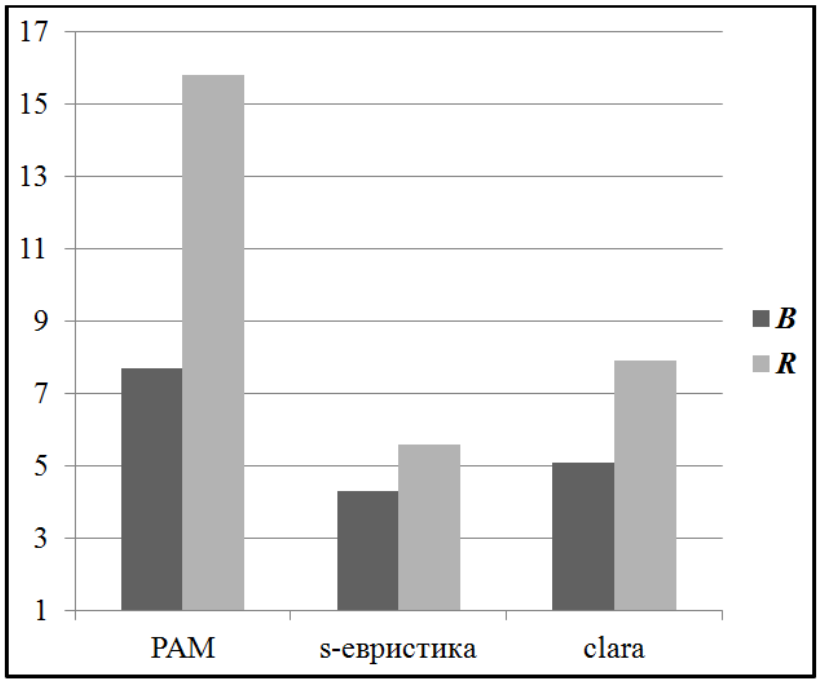

Рис. 4.4. Швидкість виконання кластеризації

Запропонований модифікований метод острівної кластеризації текстів в цілому протестовано з наступними налаштуваннями:

– для другого різновиду значення параметру k обрано рівним потроєному значенню, що використовується оригінальним методом острівної кластеризації, обчисленого за формулою (2.1);
– для третього різновиду значення параметру s обрано рівним 15.

Результати оцінювання якості отриманих результатів кластеризації з використанням оригінального методу острівної кластеризації та запропонованого модифікованого методу наведено на рис. 4.5. Отримані значення відповідають теоретичним припущенням про підвищення якості результатів кластеризації запропонованим модифікованим методом острівної кластеризації, що забезпечується різними запропонованими підходами до оброблення графу сумісної зустрічальності термів та його кластеризацією методом k-medoids.

Результати тестування швидкості виконання кластеризації запропонованим модифікованим методом острівної кластеризації текстів наведено на рис. 4.6. Вони подані у вигляді співвідношення часу виконання кластеризації модифікованим методом до часу виконання кластеризації оригінальним острівним методом. Таким чином, чим ближче до 1 значення, тим показана краща швидкість кластеризації.

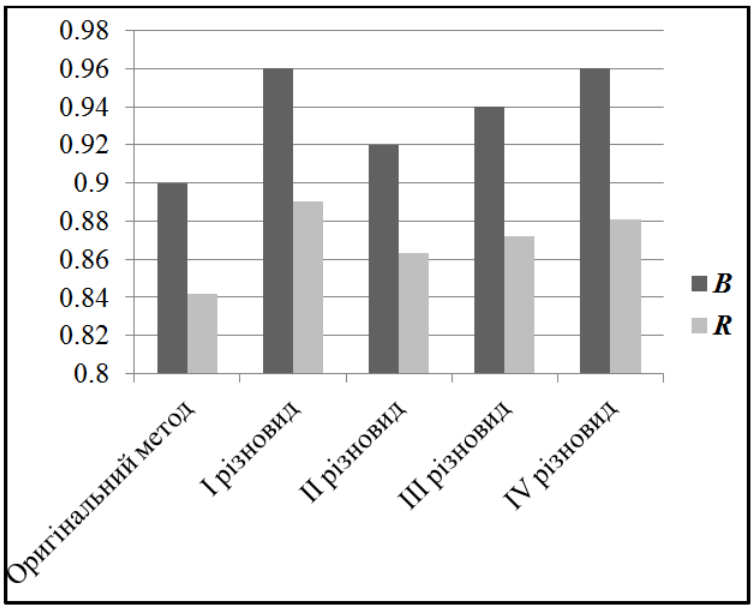

Рис. 4.5. Якість отриманих результатів кластеризації з використанням запропонованого модифікованого методу

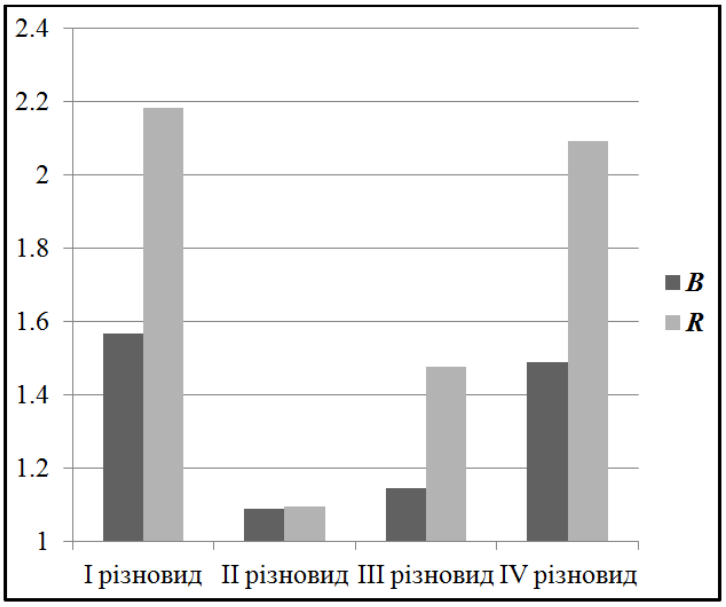

Рис. 4.6. Швидкість виконання кластеризації запропонованим модифікованим методом

Висновки за четвертим розділом

В рамках даного розділу проведено аналіз запропонованого модифікованого методу острівної кластеризації текстів, який базується на різних підходах до оброблення графу сумісної зустрічальності термів та його кластеризації методом k-medoids. Також розглянуто основні способи оцінювання методів кластеризації – оцінки якості отриманих результатів та швидкості виконання кластеризації.

Тестування проведено за допомогою розробленого в рамках даної роботи програмного забезпечення для автоматичної кластеризації текстів на двох тестових колекціях, з використанням результатів оригінального методу острівної кластеризації в якості еталону. Протестовано окремі гіпотези, які складають модифікований метод острівної кластеризації – запропоновані підходи до оброблення графу сумісної зустрічальності термів та використання методу k-medoids.

Як показало тестування, якість отриманих результатів кластеризації запропонованим модифікованим методом перевищує якість результатів, отриманих оригінальним методом, на 2-6%. Можливо зробити висновок, що запропонований модифікований метод острівної кластеризації текстів доцільно використовувати в прикладних задачах, де важливою є якість отриманих результатів, оскільки з його використанням кластеризація виконується повільніше.

5. ПОБУДОВА БІЗНЕС МОДЕЛІ

5.1. Опис проблеми

Розроблений в даній роботі метод кластеризації текстових колекцій може використовуватись в будь-якій області, де необхідний підбір документів за змістом та тематикою.

Однією з таких актуальних на сьогодні областей є підбір подібних новин на новинних веб-порталах. В цій області до сих пір існує проблема низької якості такого підбору, що пов’язано з досить великою кількістю факторів [18].

По-перше, це використання дуже простих критеріїв при виконанні підбору подібних новин. На багатьох веб-порталах новин для цього використовуються ключові слова або віднесення кожної новини до однієї або декількох наперед визначених рубрик (наприклад, економіка, політика, спорт, наука і т.д. [46]).

При цьому для використання таких критеріїв виділяється значна кількість ресурсів. Цими ресурсами виступають як і людино-години, в випадку, якщо ключові слова або рубрики необхідно задавати вручну, так і обчислювальні ресурси, якщо ці ключові слова виділяються автоматично чи автоматично визначаються рубрики. В будь-якому випадку, ці ресурси також потребують грошових затрат, що в поєднанні з низькою якістю результату призводить до вкрай неефективного їх використання.

Також одним із факторів низької якості підбору подібних новин є відсутність у переважної більшості інструментів автоматичного оновлення результатів підбору з плином часу. Пошук подібних новин відбувається лише один раз при додаванні новини, таким чином для новини, що вийшла раніше, не пропонуються новини, що вийшли після її публікації. Наприклад, якщо це перша новина про якусь подію чи явище, в добірку подібних новин будуть додані тексти, що є досить далекими за змістом і подібними лише за тематикою.

Низька якість підбору новин також впливає на складність проведення аналітики потоку новин. Адже користувачами процедури підбору є не тільки кінцеві користувачі новинного порталу, але і, наприклад, його редакція. В такому випадку через низьку якість добірок, редакції досить складно знаходити гарячі новини для головної сторінки порталу та відбирати теми для авторських редакційних матеріалів.

Описані проблеми узагальнено в дерево проблем, що зображено на рис. 5.1.

Рис. 5.1. Дерево проблем

5.2. Зацікавлені сторони

В вирішенні описаної вище проблеми прямо чи опосередковано зацікавлено досить багато різних сторін.

Є очевидним, що найбільш зацікавленими в вирішенні даної проблеми є користувачі, співробітники та власники новинного порталу, оскільки саме вони за рахунок її вирішення отримують найбільше задоволення власних потреб та інтересів.

Користувачі веб-порталу новин в першу чергу є зацікавленими в отриманні якісного контенту. Завдяки підвищенню якості добірок подібних новин, зростає загальна якість контенту, що отримується.

Співробітники порталу зацікавлені у підвищенні якості та ефективності власної роботи. Завдяки автоматичному рішенню з підбору подібних новин, зі співробітників знімається відповідальність з ручного формування таких добірок, а редакції порталу набагато простіше виконувати аналітику потоку новин.

Власник веб-порталу новин зацікавлений в зменшенні витрат та збільшенні доходів від роботи порталу. Вирішивши описану проблему, звільнені ресурси, що витрачались раніше на підбір новин, можуть бути використанні в інших напрямках роботи. Також завдяки якісним результатам підбору користувачі будуть більше часу проводити на веб-порталі, збільшуючи таким чином дохід від реклами. А якщо власником використовується така модель монетизації, як платний доступ до деяких матеріалів, то збільшується ймовірність покупки такої підписки користувачами.

Опосередковано в вирішенні описаної проблеми зацікавлені такі сторони, як рекламодавці, суспільство, державні органи і інші. Рекламодавці зацікавлені в підвищенні ефективності рекламних кампаній (а збільшення показів реклами впливає на це), суспільство – у збільшенні інформованості громадян, державні органи – у підвищенні ефективності проведення власної інформаційної політики. Проте ці зацікавлені сторони мають невеликий вплив.

Описані зацікавлені сторони узагальнено в матрицю зацікавлених осіб, що наведена в табл. 5.1.

Таблиця 5.1

Матриця зацікавлених осіб

Група зацікавлених осіб	Інтереси зацікавленої особи	Вплив	Стратегії приваблення
Користувачі	Отримання якісного контенту, підвищення якості підбору	Великий	Рекламні матеріали. Презентації продукту, демонстрації. Участь в тематичних заходах. Надання безкоштовних пробних версій. Забезпечення технічної підтримки.
Співробітники порталу	Підвищення якості та ефективності роботи	Великий	
Власник порталу	Зменшення витрат (за рахунок звільнення ресурсів) та збільшення доходів	Великий	
Рекламодавці	Підвищення ефективності рекламних кампаній завдяки збільшенню кількості показів реклами	Малий	
Суспільство	Збільшення інформованості громадян завдяки підвищенню якості підбору новин за змістом	Малий	
Державні органи	Підвищення ефективності проведення інформаційної політики	Малий	

5.3. Рішення. Основні характеристики

Поставлену проблему може вирішити спеціальний програмний продукт, який реалізує описану технологію кластеризації текстових документів. Завдяки описаному в даній роботі методу, потік новин ефективно кластеризується за своєю тематикою та змістом. Для кінцевого споживача контенту – читача новин – на перший погляд нічого не змінюється: до кожної новини на сайті новин пропонується деякий набір з подібних за змістом та тематикою новин. Проте, завдяки використанню програмного продукту, що пропонується, підвищується якість таких пропозицій. Також за допомогою нашого ПЗ читач отримує найбільш подібні за змістом новини, незважаючи на дату публікації. В наслідок чого,

якщо це перша новина про якусь подію, в добірку потраплять також новини, що з’явились пізніше і розкривають цю подію в повному обсязі.

Очевидно, що клієнтом даного програмного продукту є власне портали новин, саме тому він повинен легко інтегруватись в існуючі системи керування контентом та веб-портали в якості плагіну. Таким чином, споживачу цього програмного забезпечення достатньо лише один раз інсталювати та налаштувати його взаємодію з існуючою системою. При цьому подальша робота клієнту з власним веб-порталом не змінюється: програмний продукт надалі працює автономно.

При публікації нової новини через систему керування контентом програмне забезпечення запускається автоматично та виконує кластеризацію. При запиті на перегляд певної новини до самого порталу новин, програмний продукт виконає додавання посилань на подібні новини, звернувшись до збережених результатів кластеризації. Таким чином, добірки подібних новин оновлюються кожний раз при додаванні нової новини.

5.4. Конкурентні переваги рішення

Конкурентів даного програмного рішення на ринку досить багато. Умовно їх можливо розділити на дві великі групи:

- плагіни для різноманітних систем керування контентом;
- унікальні власні рішення, розроблені кожним порталом самостійно для своїх потреб.

Наприклад, для WordPress (одна з популярних систем керування контентом [47]) лише на одному з маркетів знайдено близько 50 різноманітних плагінів для підбору подібних записів. Проте всі з них працюють на основі ключових слів, які необхідно проставляти вручну, або на основі категорій, до яких віднесено запис. Є очевидним те, що для отримання якісного результату за допомогою такого ПЗ необхідно докласти багато людських зусиль.

Оцінити можливості та особливості програмних продуктів з другої категорії досить важко, оскільки всі з них є внутрішніми розробками великих новинних порталів та служб новин. Проте можливо проаналізувати результати їх роботи, вивчивши результати підбору подібних новин, і на основі цього зробити певні висновки. В перелік служб новин, чиї веб-портали було проаналізовано, потрапили такі відомі та досить великі служби, як ТСН, BBC і інші. Визначити, чи створюються добірки подібних новин на цих сайтах автоматично чи вручну не вдалося, проте всі вони мають недоліки. Наприклад, таким недоліком є відсутність оновлення добірок.

Отже, конкурентними перевагами програмного продукту, що пропонується, є:

- висока якість підбору подібних новин за рахунок використання кластеризації;
- автоматичне оновлення результатів підбору з плином часу;
- мінімізація ресурсів, які витрачаються на підбір – як людських, так і обчислювальних;
- легкість інтегрування програмного продукту в веб-портали.

5.5. Клієнти. Сегменти ринку споживання

Як вже зазначалось вище, потенційними клієнтами продукту, що пропонується, є новинні веб-портали. Проте через їх значну кількість та різноманітність необхідне проведення їх сегментації, з метою вибору ринків для виходу продукту. Наприклад, за даними bigmir.net, лише в Україні до категорії «ЗМІ та періодика» віднесено 4278 веб-сайтів [48]. При цьому до цієї категорії не було віднесено спеціалізовані новинні сайти, які підпадали під інші категорії – наприклад «Спорт», «Література», «Кіно» і так далі. Є очевидним, що запропонований продукт не може бути однаково привабливий для всіх цих сайтів і однаково для них

позиціонуватись. Сегментацію ринку споживання проведено за декількома критеріями.

В першу чергу ринок новинних веб-порталів сегментовано за мовою розміщених матеріалів:

- англійська;
- німецька;
- французька;
- іспанська;
- російська і т.д..

Необхідність в проведенні сегментації за мовою, крім маркетингових причин, також має технологічний фактор. Це пов’язано з тим, що незважаючи на те, що описаний в роботі метод кластеризації не має мовних обмежень, будь-які його програмні реалізації потребують словники для кожної мови. Наприклад, якість та повнота таких словників для української мови є недостатньою для її підтримки продуктом, що описується.

За типом порталу ми можемо виділити такі сегменти:

- спеціалізовані служби новин;
- новинні агрегатори;
- персональні блоги.

За кількістю відвідувачів:

- малі (до 3000 унікальних відвідувачів за добу);
- середні (від 3000 до 15000 унікальних відвідувачів за добу);
- великі (від 15000 унікальних відвідувачів за добу).

Виділивши критерії та сегменти, що їм відповідають, обрано ті сегменти, що є найбільш привабливими для просування продукту.

Найбільш перспективними є англомовні середні та великі спеціалізовані служби новин, а також англомовні персональні блоги. Переважна більшість блогів працює на одній з декількох найбільш популярних систем керування контентом, тому розробивши для них

спеціалізовані версії програмного продукту, можливо їх розмістити на торгових майданчиках цих CMS.

Сегмент середніх та великих спеціалізованих служб новин є більш перспективним з точки зору прибутків, проте для роботи з ним необхідно прикласти більше зусиль. Наприклад, це пов’язано з тим, що для переважної більшості клієнтів з цього сегменту буде необхідно розроблювати окремі інтеграційні модулі, оскільки такі веб-портали використовують власні системи керування контентом.

Саме через необхідність додаткових витрат, сегмент малих спеціалізованих служб новин є мало перспективним. Також мало перспективним є сегмент новинних агрегаторів, оскільки вони не є зацікавленими в продукті і пропозиції подібних новин.

5.6. Унікальна ціннісна пропозиція

Ціннісна пропозиція описує ті переваги, які надають споживачу наші товари та послуги і які вирішують проблеми споживачів [49]. Відповідно, унікальною буде така пропозиція, яка чітко відділяє товар від конкурентів і дає зрозуміти, чому необхідно придбати саме наш товар.

Для кожного сегменту споживачів зазвичай пропонується виділяти окрему ціннісну пропозицію [50].

Ціннісною пропозицією для сегменту персональних блогів, в рамках продукту, що пропонується, є повністю автоматичний підбір подібних записів в блозі. Оскільки всі інші продукти для даного сегменту, крім інсталювання плагіну для CMS, потребують також додаткових дій при кожному додаванні запису в блог (виділити ключові слова, розмістити запис в певних рубриках і так далі), то наш продукт працює повністю автоматично. Таким чином, ця ціннісна пропозиція також є унікальною.

Для сегменту середніх та великих новинних порталів акцент ціннісної пропозиції зміщується з повної автоматизації на збільшення якості підбору та збільшення прибутку новиного порталу завдяки цьому.

Адже з якісними добірками подібних новин збільшується ймовірність того, що користувачі перейдуть з певної конкретної новини на подібну і так далі. Середній час, який на сайті проводить користувач, збільшується, а це в свою чергу збільшує кількість переглядів сайту, кількість показів реклами, а також створює можливість отримання більш вигідних рекламних контрактів.

5.7. Доходи та витрати

Сумарний дохід складається як сума доходів від продажу товарів та супутніх послуг для кожного сегменту споживачів.

Для сегменту персональних блогів пропонується продавати наступні товари та надавати наступні послуги:

- плагіни для популярних CMS (WordPress, Joomla, Drupal і так далі), що реалізовують підбір подібних записів з застосуванням описаного методу кластеризації;
- безкоштовні версії даних плагінів з певними обмеженнями (кількість записів всього, що оброблюється, обмежена кількість записів в добірках тощо) та показом реклами;
- надання розширеної платної технічної підтримки.

Розглянуті плагіни розміщуються на популярних біржах з їх продажі для відповідних CMS. Ціна такого програмного продукту для кінцевого споживача складає приблизно 10 доларів США, з яких деякий відсоток отримує сама біржа. Цей відсоток залежить від умов конкретного майданчика, тому, на далі, для розрахунків буде використовуватись середнє значення в 10%. Також дана платна ліцензія включає отримання всіх оновлень з виправленнями помилок протягом одного календарного року з дня покупки.

Розширена платна технічна підтримка також надається на один календарний рік та включає в себе гарантоване оброблення запитів протягом 24-х годин в робочі дні.

Для сегменту середніх та великих спеціалізованих служб новин основним продуктом є модуль підбору подібних новин, що реалізовує описаний метод кластеризації. Продаж даного продукту також включає в себе послуги з інтеграції цього модуля в систему керування контентом клієнта, якщо необхідно. Ліцензія, що надається разом з цим продуктом, дає можливість протягом року отримувати всі оновлення продукту, а не лише ті, що виправляють помилки. Технічна підтримка клієнту надається в розширеному обсязі – гарантоване оброблення запитів протягом 24-х годин.

Розрахований дохід по місяцям протягом першого року наведено в табл. 5.2. Всі суми наведено в доларах США.

Таблиця 5.2

Доходи

	Реклама	Продаж плагінів	Розширена технічна підтримка	Продаж модулів для служб новин	Всього
1	10	270	0	0	280
2	15	2970	90	0	3075
3	25	4050	130	0	4205
4	40	4770	150	0	4960
5	55	5670	180	0	5905
6	70	6390	210	0	6670
7	80	6840	220	200	7340
8	90	7290	240	400	8020
9	100	7920	260	400	8680
10	110	8370	270	600	9350
11	115	8820	290	900	10125
12	120	9000	300	1300	10720

Сукупність загальних витрат складають наступні витрати:

– витрати на розроблення, підтримку та вдосконалення програмних продуктів (утримання робочих місць, придбання необхідних інструментів та програмних засобів);
– витрати на надання послуг технічної підтримки (утримання робочих місць, придбання необхідних інструментів та програмних засобів);
– витрати на оплату праці (заробітна плата та інші виплати працівникам);
– витрати на опалення, освітлення, водопостачання, водовідвід;
– адміністративні витрати (витрати на зв'язок, податки та збори, плата за послуги банків тощо);
– витрати на рекламу та дослідження ринку.

Витрати по місяцям протягом першого року наведено в табл. 5.3. Всі суми наведено в доларах США.

Таблиця 5.3

Витрати

	Розроблення	Технічна підтримка	Оплата праці	Комунальні та адміністративні	Реклама	Всього
1	4000	1000	2500	500	300	8300
2	130	50	2500	500	400	3580
3	130	50	2500	500	500	3680
4	130	50	2500	500	700	3880
5	130	50	2500	500	800	3980
6	130	50	2500	500	800	3980
7	130	50	2500	500	800	3980
8	130	50	2500	500	800	3980
9	130	50	2500	500	800	3980
10	130	50	2500	500	800	3980
11	130	50	2500	500	800	3980

Продовження табл. 5.3

12	130	50	2500	500	800	3980

Розрахуємо маржинальний прибуток за перший рік за формулою (5.1) [51].

$$\text{Маржинальний прибуток} = \text{Дохід} - \text{Витрати} \quad (5.1)$$

За формулою (5.1) маржинальний прибуток за перший рік складе 28050 доларів США.

5.8. Бізнес-модель

Узагальнимо вище наведену інформацію в виглядів блоків, що складають так звану канву бізнес-моделі або lean canvas [52].

Споживачі:

- ранні клієнти: персональні блоги;
- середні та великі спеціалізовані служби новин.

Проблема:

- низька якість підбору подібних новин;
- значне використання ресурсів існуючими рішеннями;
- відсутність автоматичного оновлення добірок.

Рішення:

- програмний продукт, що автоматично та якісно формує добірки подібних новин.

Унікальна ціннісна пропозиція:

- повністю автоматичне рішення, яке працює за принципом, що можливо сформулювати як «Інсталювати та забути»;
- можливість більше заробляти завдяки якісним добіркам.

Потоки доходів:

- продаж плагінів для блогів на популярних системах керування контентом;

– дохід від реклами в безкоштовних версіях даних плагінів;
– продаж послуг з розширеної технічної підтримки;
– продаж модулю підбору подібних новин для спеціалізованих веб-порталів.

Структура витрат:

– витрати на розроблення, підтримку та вдосконалення програмних продуктів;
– витрати на надання послуг технічної підтримки;
– витрати на оплату праці;
– витрати на опалення, освітлення, водопостачання, водовідвід;
– адміністративні витрати;
– витрати на рекламу та дослідження ринку.

Також в канву бізнес-моделі включаються структурні блоки, які ще не були розглянуті. Це прихована перевага (перевага, яку не можливо скопіювати або купити), ключові метрики (основні показники, що вимірюються) та канали (шляхи до користувачів).

В якості каналів контакту з клієнтами пропонується використовувати наступні шляхи: SEO, SMM, розміщення продукту на спеціальних торгівельних майданчиках, прямі контакти для продаж.

Прихованою перевагою нашого продукту виступає технологія підбору подібних новин на основі кластеризації.

Ключовими метриками є наступні: кількість встановлень безкоштовних версій, кількість продаж платних версій, кількість оновлень до наступних версій, кількість показів реклами в безкоштовних версіях, кількість запитів в службу технічної підтримки.

Описані структурні блоки об’єднано в єдину канву бізнес-моделі, котру наведено в додатку 2.

Висновки за п’ятим розділом

В даному розділі розглянуто питання практичного застосування розробленого методу кластеризації текстових колекцій.

Незважаючи на те, що побудована бізнес-модель потребує доопрацювань, вона підтверджує життєздатність стартапу, заснованого на застосуванні технології кластеризації для підбору подібних новин. Описані сегменти споживачів, які готові платити за вирішення даної проблеми, наявні унікальні ціннісні пропозиції для кожного сегменту. Запропонований програмний продукт має конкурентні переваги.

Розраховані доходи та витрати протягом першого року роботи. Знайдений маржинальний прибуток за цей період – 28 тисяч доларів США – також підтверджує життєздатність описаної бізнес-моделі.

ВИСНОВКИ

У даній роботі запропоновано модифікований метод острівної кластеризації природномовних текстових даних. Даний метод є ефективним у разі необхідності виконання кластеризації текстів будь-якої тематики на будь-якій природній мові (що забезпечується необхідністю лише мінімального набору лінгвістичних ресурсів, які використовуються лише для попереднього оброблення текстів).

Докладно описана постановка задачі кластеризації природномовних текстових даних у загальному вигляді. Проаналізовано існуючі методи кластеризації текстів, описані їх переваги та недоліки. На основі аналізу існуючих методів поставлено вимоги, що визначають ефективність методу кластеризації, яким розроблюваний метод повинен відповідати в повній мірі. Враховуючи ці вимоги, обрано метод острівної кластеризації для його подальшої модифікації та вдосконалення в якості основи для розроблюваного методу. Для модифікації обрано етапи попереднього оброблення графу сумісної зустрічальності термів та кластеризації отриманого після оброблення наближення графу.

Запропоновано нові підходи до попереднього оброблення графу сумісної зустрічальності термів, запропоновано та проаналізовано використання методу k-medoids для кластеризації отриманого після оброблення наближення графу. Запропоновано модифікований метод острівної кластеризації природномовних текстових даних, що використовує запропоновані підходи та метод k-medoids.

Для тестування запропонованого модифікованого методу острівної кластеризації природномовних текстових даних розроблено програмне забезпечення для автоматичної кластеризації текстів. Розроблене програмне забезпечення містить реалізації алгоритмів оригінального методу острівної кластеризації та запропонованого модифікованого методу в різних різновидах.

Проведено аналіз ефективності запропонованого модифікованого методу острівної кластеризації за допомогою обчислення показника якості отриманих результатів кластеризації та швидкості її виконання. Як показало тестування, запропонований модифікований метод острівної кластеризації текстів забезпечує вищу якість результатів, ніж оригінальний метод острівної кластеризації, та його доцільно використовувати в прикладних задачах, де важливою є якість отриманих результатів, оскільки з його використанням кластеризація виконується повільніше.

Також в рамках даної роботи розроблено бібліотеку оброблення природномовних текстів SimpleNetNlp, яка має відкритий програмний код і доступна для завантаження в системі керування пакетами Nuget.

Пріоритетними подальшими напрямками дослідження є аналіз способів оптимізації розробленого методу з метою підвищення його швидкодії та розроблення чітких методик з вибору конкретного різновиду запропонованого методу в залежності від галузі його практичного застосування.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Information explosion [Електронний ресурс]. – Режим доступу: https://en.oxforddictionaries.com/definition/information_explosion. – Назва з екрану. – (Дата звернення: 15.12.2017).
2. Gantz J., Reinsel D. The digital universe in 2020: Big data bigger digital shadows and biggest growth in the far east // IDC iView: IDC Anal. Future. – 2012. – №2007. – С.1-16.
3. Berry M.W. Survey of Text Min-ing // Springer. – 2003.
4. Salton G., Wong A., and Yang C.S. A Vector Space Model for Automatic Indexing // Communications of the ACM. – №18. – С.613-620.
5. Афонин А.А., Крейнес М.Г. Кластеризация текстовых коллекций: помощь при содержательном поиске и аналитический інструмент // сб. науч. ст. "Интернет-порталы: содержание и технологии". – Москва: ФГУ ГНИЙ ИТТ Информика. – 2007. – №4.
6. Stefanowski J., Weiss D. Comprehensible and Accurate Cluster Labels in Text Clustering // Dawid Weiss Institute of Computing Science. – 2001.
7. Hotho A., Staab S., Stumme G. Explaining text clustering results using semantic structures // Principles of Data Mining and Knowledge Discovery, 7th European Conference, PKDD. – 2003.
8. Bharati A., Varanasi K., Kamisetti C., Sangal R., Bendre S. A Document Space Model for Automated Text Classification based on Frequency Distribution across Categories // HIT TechReport. – 2002. – №6.
9. Quinlan J. C4.5: Programs for Machine Learning // Morgan Kaufman. – 1993.
10. Мисуно И.С., Рачковский Д.А., Слипченко СВ., Соколов А.М. Поиск текстовой информации с помощью текстовых представлений // Проблеми програмування. – 2005. – №4. – С.50-59.
11. Губин М. Модели и методы представления текстового документа в системах информационного поиска : дисс. к.ф.-м.н.. – СПГУ, 2005.

12. van Rijsbergen C.J. Information Retrieval //. – Лондон. – 1979.
13. Ravin Y., Leacock C. Polysemy: Theoretical and Computational Approaches //. – New York: Oxford University Press. – 2000.
14. Hartigan J., Wong M. A K-Means Clustering Algorithm // Applied Statistics. – 1979. – №1. – С.100-108.
15. Cutting D.R., Pedersen J.O., Karger D., and Tukey J.W. Scatter/gather: A cluster-based approach to browsing large document collections // Proceedings of 15th Annual ACM-SIGIR. – 1992. – С.318-329.
16. Hofmann T. Probabilistic Latent Semantic Indexing // Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval. – 1999. – С.50-55.
17. Ukkonen E. On-line construction of suffix trees // Algorithmica. – 1995. – №3. – С.249-260.
18. Шмулевич М. М. Метод автоматической кластеризации текстов, основанный на извлечении из текстов имен объектов и последующем построении графов совместной встречаемости ключевых термов : дис. канд. фіз.-мат. наук / Шмулевич Марк Михайлович – МФТИ, 2009.
19. Spielman D.A., Srivastava N. Graph sparsification by effective resistances // Symposium on Theory of Computing 2004. – 2004. – С.81-90.
20. Ghosh A., Boyd S., Saberi A. Minimizing effective resistance of a graph // 17th International Symposium on Mathemati-cal Theory of Networks and Systems. – 2006. – С.1185-1196.
21. Ovelgonne M. Scalable Algorithms for Community Detection in Very Large Graphs. – 2011.
22. Park H., Jun C. A simple and fast algorithm for K-medoids clustering // Expert Systems with Applications. – 2009. – №36. – С.3336-3341.
23. Richter J. CLR via C# 4th Edition // Microsoft Press. – 2012.
24. .NET Framework [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/.NET_Framework. – Назва з екрану. – (Дата звернення: 01.02.2018).

25. TIOBE Index [Електронний ресурс]. – Режим доступу: https://www.tiobe.com/tiobe-index/. – Назва з екрану. – (Дата звернення: 01.02.2018).
26. C# [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/C_Sharp. – Назва з екрану. – (Дата звернення: 01.02.2018).
27. What’s New in Visual Studio 2015 [Електронний ресурс]. – Режим доступу: https://msdn.microsoft.com/en-us/library/bb386063.aspx. – Назва з екрану. – (Дата звернення: 01.02.2018).
28. Git [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Git. – Назва з екрану. – (Дата звернення: 01.02.2018).
29. Git [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Git. – Назва з екрану. – (Дата звернення: 01.02.2018).
30. Stanford CoreNLP – Natural language software [Електронний ресурс]. – Режим доступу: https://stanfordnlp.github.io/CoreNLP/. – Назва з екрану. – (Дата звернення: 01.02.2018).
31. The Stanford CoreNLP Natural Language Processing Toolkit / [C. D. Manning, M. Surdeanu, J. Bauer та ін.] // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations / [C. D. Manning, M. Surdeanu, J. Bauer та ін.]., 2014. – С. 55–60.
32. The Stanford NLP Software for .NET! [Електронний ресурс]. – Режим доступу: http://sergey-tihon.github.io/Stanford.NLP.NET/. – Назва з екрану. – (Дата звернення: 01.02.2018).
33. SimpleNetNlp [Електронний ре-сурс]. – Режим доступу: https://github.com/yakivyusin/SimpleNetNlp. – Назва з екрану. – (Дата звернення: 15.11.2017).

34. Способи прискорення обчислення матриці кореляції термів в методі острівної кластеризації текстів / Юсин Я.О., Заболотня Т.М. // Прикладна математика та комп'ютинг ПМК-2018 : Тез. доп. – Київ, 21-23 березня 2018. – С.272–276.
35. Pfitzner D., Leibbrandt R., Powers D. Characterization and evaluation of similarity measures for pairs of clusterings // Knowledge and Information Systems. – 2009. – №19. – С.361–394.
36. Feldman R., Sanger J. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. – 2007.
37. Manning C. D., Raghavan P., Schütze, H. Introduction to Information Retrieval. – 2009.
38. Estivill-Castro V. Why so many clustering algorithms — A Position Paper // ACM SIGKDD Explorations Newsletter. – 2002. – №1. – С.65–75.
39. Davies D.L., Bouldin D.W. A cluster separation measure // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1979. – №1.
40. Dunn J.C. Well separated clusters and optimal fuzzy-partitions // Journal of Cybernetics. – 1974. – №4.
41. Kaufman L., Rousseeuw P. Finding Groups in Data. An Introduction to Cluster Analysis. – 2005.
42. Hruschka E.R., Vendramin L., Campello R.J. G.B. On the comparison of relative clustering validity criteria // 2009 SIAM International Conference on Data Mining. – 2009. – №1.
43. Rand, W. M. Objective criteria for the evaluation of clustering methods // Journal of the American Statistical Association. – 1971. – №66. – С.846–850.
44. BBC News [Електронний ресурс]. – Режим доступу: http://www.bbc.com/news. – Назва з екрану. – (Дата звернення: 01.03.2018).

45. Reuters-21578 [Електронний ресурс]. – Режим доступу: http://www.daviddlewis.com/resources/testcollections/reuters21578/. – Назва з екрану. – (Дата звернення: 01.03.2018).
46. The Most Popular Blog Categories According To Google [Chart] [Електронний ресурс]. – Режим доступу до ресурсу: https://goo.gl/FhtfYb. – Назва з екрану. – (Дата звернення 20.11.2017).
47. CMS Usage Statistics [Електронний ресурс]. – Режим доступу до ресурсу: https://goo.gl/rjcHM3. – Назва з екрану. – (Дата звернення 20.11.2017).
48. Главная – Рейтинг сайтов – bigmir)net [Електронний ресурс]. – Режим доступу до ресурсу: https://goo.gl/bxgzcG. – Назва з екрану. – (Дата звернення 20.11.2017).
49. Value Proposition Canvas Template [Електронний ресурс]. – Режим доступу до ресурсу: https://goo.gl/MbhgG4. – Назва з екрану. – (Дата звернення 20.11.2017).
50. Шаблон — GN1403: Моделирование бизнес-процессов [Електронний ресурс]. – Режим доступу до ресурсу: https://goo.gl/43VDek. – Назва з екрану. – (Дата звернення 20.11.2017).
51. Маржинальная прибыль. Пример анализа. Формула расчета [Електронний ресурс]. – Режим доступу до ресурсу: https://goo.gl/kyFxfH. – Назва з екрану. – (Дата звернення 20.11.2017).
52. Как правильно заполнять Lean Canvas | Wiki.Rademade.com [Електронний ресурс]. – Режим доступу до ресурсу: https://goo.gl/Gkb4Uq. – Назва з екрану. – (Дата звернення 20.11.2017).

ДОДАТКИ

Додаток 1
Список стоп-слів

a
about
above
across
after
afterwards
again
against
all
almost
alone
along
already
also
although
always
am
among
amongst
amoungst
amount
an
and
another
any
anyhow
anyone
anything
anyway
anywhere
are
around
as
at
back
be
became
because
become
becomes
becoming
been
before
beforehand
behind
being
below
beside
besides
between
beyond
bill
both
bottom
but
by
call
can
cannot
cant
co
con
could
couldnt
cry
de
describe
detail
do
done
down
due
during
each
eg
eight
either
eleven
else
elsewhere
empty
enough
etc
even
ever
every
everyone
everything
everywhere
except
few
fifteen
fify
fill
find
fire
first
five
for
former
formerly
forty
found
four
from
front
full
further
get
give
go
had
has
hasnt
have
he
hence
her
here
hereafter
hereby
herein
hereupon
hers
herself
him
himself
his
how
however
hundred
i
ie
if
in
inc
indeed
interest
into
is
it
its
itself
keep
last
latter
latterly
least
less
ltd
made
many
may
me
meanwhile
might
mill
mine
more
moreover
most
mostly
move
much
must
my
myself
name
namely
neither
never
nevertheles
s
next
nine
no
nobody
none
noone
nor
not
nothing
now
nowhere
of
off
often
on
once
one
only
onto
or
other
others
otherwise
our
ours
ourselves
out
over
own
part
per
perhaps
please
put
rather
re
s
same
see
seem
seemed
seeming
seems
serious
several
she
should
show
side
since
sincere
six
sixty
so
some
somehow
someone
something
sometime
sometimes
somewhere
still
such
system
take
ten
than
that
the
their
them
themselves
then
thence
there
thereafter
thereby
therefore
therein
thereupon
these
they
thick
thin
third
this
those
though
three
through
throughout
thru
thus
to
together
too
top
toward
towards
twelve
twenty
two
un
under
until
up
upon
us
very
via
was
we
well
were
what
whatever
when
whence
whenever
where
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
whoever
whole
whom
whose
why
will
with
within
without
would
yet
you
your
yours
yourself
yourselves
's

Додаток 2
Канва бізнес-моделі

<table>
<tr>
<td rowspan="2">Проблема
Низька якість підбору подібних новин.

Відсутність автоматичного оновлення.

Значне використання ресурсів.</td>
<td>Рішення
Програмний продукт, що автоматично та якісно формує добірки.</td>
<td rowspan="2">Унікальна ціннісна пропозиція
Інсталювати та забути.

Більше заробляй з якісними добірками.</td>
<td>Прихована перевага
Технологія підбору подібних новин на основі кластеризації.</td>
<td rowspan="2">Споживачі
Ранні клієнти: персональні блоги.

Середні та великі портали новин.</td>
</tr>
<tr>
<td>Ключові метрики
Кількість встановлень free версії.
Кількість продаж.
Кількість оновлень.
Кількість показів реклами.
Кількість запитів за підтримкою.</td>
<td>Канали
SEO

SMM

Прямі контакти

Торгівельні майданчики</td>
</tr>
<tr>
<td colspan="3">Структура витрат
Витрати на розроблення, підтримку та вдосконалення продуктів.
Витрати на надання послуг технічної підтримки.
Оплата праці.
Комунальні послуги.
Адміністративні витрати.
Витрати на рекламу та дослідження ринку.</td>
<td colspan="2">Потоки доходів
Продаж плагінів для блогів.
Дохід від реклами.
Продаж розширеної технічної підтримки.
Продаж модулю підбору для порталів новин.</td>
</tr>
</table>

Додаток 3
Копія презентації